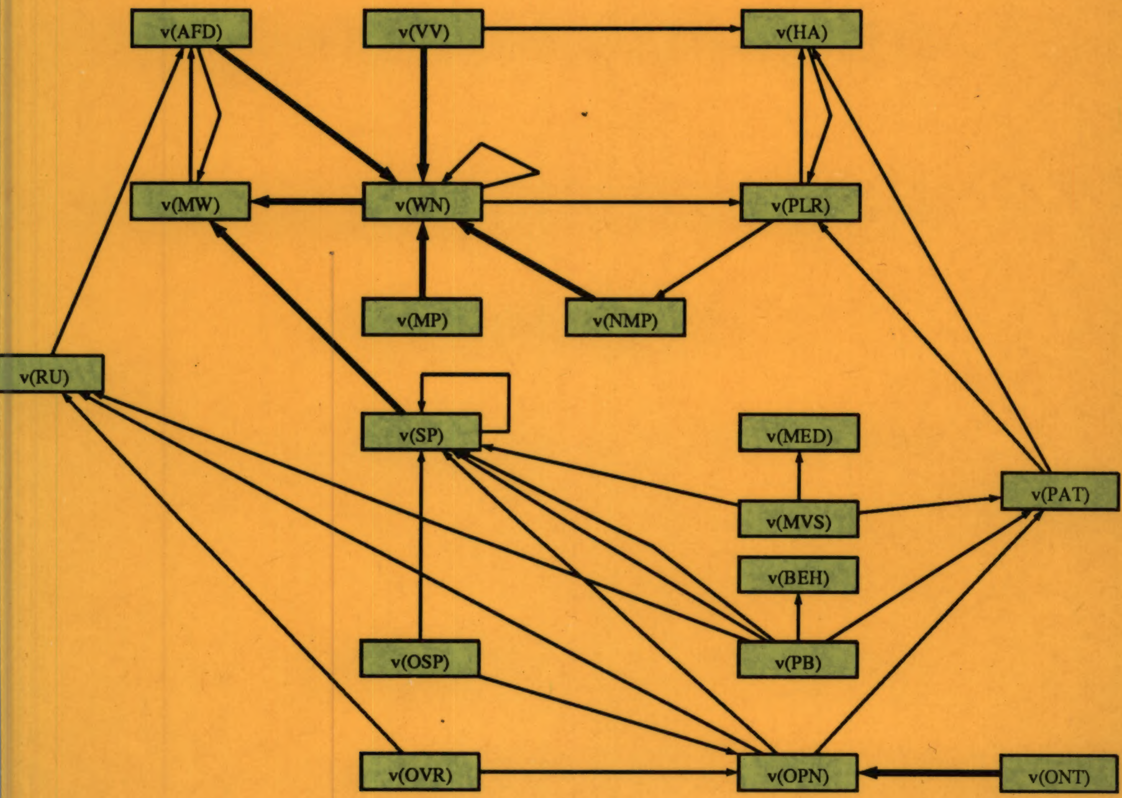
E.O. de Brock

De grondslagen van semantische databases



ACADEMIC SERVICE

De grondslagen van semantische databases

Voor Audrey
en Cherelien

E.O. de Brock

De grondslagen van semantische databases

ACADEMIC SERVICE

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Brock, E.O. de

De grondslagen van semantische databases / E.O. de Brock. -
Schoonhoven : Academic Service. - Ill.
Met index, lit. opg.
ISBN 90-6233-333-8
SISO 521.2 SVS 8.12.3 UDC 681.3.02 NUGI 852
Trefw.: databanken.

10 9 8 7 6 5 4 3 2 1

Uitgegeven door: Academic Service
Postbus 81
2870 AB Schoonhoven

Zetwerk: Dit boek is door de auteur camera-ready aangeleverd.
Omslagontwerp: FennArt, Steenwijk
Druk: Krips Repro Meppel
Bindwerk: Meeuwis, Amsterdam

ISBN 90 6233 333 8
NUGI 852

INHOUD

VOORWOORD

De complexiteit van informatiesystemen neemt in de praktijk hand over hand toe. De behoefte aan formele specificaties van zulke complexe systemen wordt dientengevolge steeds groter. Deze behoefte betreft niet alleen de formele specificatie van het zogeheten *gegevensmodel* van een organisatie, maar ook van het zogeheten *procesmodel.* Het gegevensmodel bestaat uit de voor deze organisatie relevante gegevens(structuren) en de voor deze gegevens geldende randvoorwaarden (die in dit verband meestal *constraints* worden genoemd). Het procesmodel daarentegen bevat de voor de organisatie relevante raadpleeg- en wijzigingsoperaties op de gegevens. Deze formele modellen kunnen als leidraad dienen voor de implementatie en het gebruik van dergelijke complexe systemen. Met de huidige, op de markt beschikbare database-managementsystemen laten zulke wiskundige modellen zich overigens op steeds directere wijze naar een implementatie vertalen. Deze tendens dat de afstand tussen formele specificatie en implementatie steeds kleiner wordt, onderstreept eens te meer het groeiende belang van formele specificaties van informatiesystemen.

Dit boek laat zien hoe dergelijke (systeemonafhankelijke) formele specificaties er uit kunnen zien en ook in complexe situaties praktisch bruikbaar (en in feite zelfs onmisbaar) zijn. Tevens wordt duidelijk gemaakt hoe een belangrijk deel van de *semantiek* van de gegevens kan worden weerspiegeld *in* het formele gegevens- en procesmodel van een organisatie. Hieraan ligt het inzicht ten grondslag dat de semantiek van de gegevens bepalend is voor de constraints waaraan deze gegevens en de operaties op deze gegevens dienen te voldoen. Hoe deze door de semantiek bepaalde constraints als wezenlijk onderdeel in de definitie van het wiskundige model van de gegevens en hun operaties zelf kunnen worden opgenomen, is één van de hoofdthema's van dit boek. Omdat het juist de intentie van semantische dababases is om zoveel mogelijk van de semantiek van de gegevens in het formele model zelf op te nemen, is hiermee tevens de titel van het onderhavige boek verklaard. De hier gegeven aanpak is reeds herhaalde malen in zeer uiteenlopende praktijksituaties toegepast en heeft daarbij zijn bruikbaarheid voor het modelleren van informatiesystemen duidelijk bewezen.

Het onderhavige boek is een grondige bewerking en uitbreiding van eerdere aantekeningen ten behoeve van een college Database-systemen dat ondergetekende de afgelopen vijf jaar heeft gegeven aan de Technische Universiteit Eindhoven.

Hoofdstuk 0 bevat de definities en elementaire eigenschappen van de algemene wiskundige basisbegrippen die in de rest van het boek worden gebruikt.

In Hoofdstuk 1 worden de drie kernbegrippen *tabel, database-toestand* en *database-universum* formeel gedefinieerd. Deze begrippen vormen de basis voor de rest van stof.

Hoofdstuk 2 behandelt diverse centrale onderwerpen uit de theorie van databases:

- operaties op tabellen (projectie, join, renaming, etcetera),
- afhankelijkheden, (minimale) sleutels en normaalvormen,
- referentiële integriteit en database-functies, en
- operaties op database-universa.

Veel nadruk wordt hierbij gelegd op het belangrijke onderscheid tussen *incidentele* eigenschappen (dat wil zeggen eigenschappen van toestanden) en *structurele* eigenschappen (dat wil zeggen eigenschappen van toestandsruimten). Helaas worden in vele theoretische én praktische verhandelingen deze verschillende niveaus niet duidelijk onderscheiden.

Hoofdstuk 3 is gewijd aan het systematisch construeren van database-universa in de praktijk. In dit hoofdstuk wordt een classificatie van (statische) constraints gegeven, geïllustreerd aan de hand van diverse praktijkvoorbeelden. Ook komt hier het modelleren van *generalisatie* en *specialisatie* (alias *differentiatie*) aan bod.

Om enig inzicht te geven in de specifieke problemen die kunnen ontstaan bij complexe databases, bevat Hoofdstuk 4 een omvangrijk voorbeeld van een database-universum. Dit voorbeeld is de bekende ziekenhuis-case die intussen zowel nationaal als internationaal wordt gebruikt voor de evaluatie van de functionele eigenschappen van diverse toonaangevende database-managementsystemen, zie bijvoorbeeld [IDT 88] en [NGI 88]. Vele soorten constraints die men in de praktijk kan tegenkomen, hebben een analogon in dit niet-triviale voorbeeld.

Hoofdstuk 5 geeft een formele behandeling van *dynamische* constraints, dat wil zeggen eisen aan de toegestane toestands*overgangen*. Dit hoofdstuk bevat tevens een uitgebreid voorbeeld met vele soorten dynamische constraints, gebaseerd op het ziekenhuisvoorbeeld uit Hoofdstuk 4.

Hoofdstuk 6 gaat in op het *raadplegen* van een database. In dit hoofdstuk worden formele definities van de begrippen *query* en *view* gegeven. Ook worden deze begrippen geïllustreerd aan de hand van diverse voorbeelden.

Hoofdstuk 7 bevat een wiskundige behandeling van het *onderhoud* van een database middels elementaire tot zeer complexe (trans)acties. De transacties worden hierbij dan ook niet imperatief of operationeel maar declaratief beschreven. Naast het algemene transactiebegrip worden ook enige meer specifieke, SQL-achtige klassen van transacties gedefinieerd.

In Hoofdstuk 8 wordt ingegaan op *data dictionaries*. Zoals een formele theorie van databases nodig is om te abstraheren van specifieke applicaties, zo is een formele theorie van data dictionaries nodig om te kunnen abstraheren van specifieke data dictionarysystemen. Ondanks de snel groeiende belangstelling voor (en mogelijkheden van) de huidige data dictionarysystemen schijnt er nog nauwelijks een formele theorie van data dictionaries te bestaan. In het laatste hoofdstuk van het boek wordt een begin gemaakt met zo'n formele theorie van data dictionaries. In het bijzonder wordt het data-definitiegedeelte van data dictionaries aan een nadere (formele) beschouwing onderworpen. Ter illustratie worden enige simpele structuren voor data dictionaries gedefinieerd en aan de hand van een voorbeeld helemaal "doorgerekend".

Het boek bevat een ruime sortering opgaven. Hierin komen zowel theoretische als praktische aspecten aan bod. Voor een goede verwerking van de stof is het ten zeerste aan te bevelen (een groot deel van) de opgaven ook daadwerkelijk te maken.

In het onderhavige boek wordt veel aandacht besteed aan het kunnen vertalen van informele omschrijvingen naar formele specificaties, en omgekeerd. Immers, dit is de schakel tussen de organisatie- en materie-deskundigen van de te modelleren toepassing enerzijds en de systeemdeskundigen anderzijds (waarbij het uiteraard niet uitgesloten is dat deze verschillende functies in dezelfde persoon verenigd kunnen zijn).

Langs deze weg wil ik graag alle collega's en studenten bedanken die de afgelopen jaren bewust of onbewust hebben bijgedragen aan de vele ervaringen waarop de inhoud van dit boek voor een niet gering deel berust. In het bijzonder wil ik in dit verband Frans Remmen noemen. Het evenwicht tussen theorie en praktijk dat ik in mijn werk op het gebied van databases gevonden hoop te hebben is namelijk voor een groot deel aan hem te danken.

Mijn werkgever Philips dank ik voor de geboden gelegenheid om ook een deel van het boek onder werktijd te kunnen schrijven.

Frank Pieper en Frans Remmen wil ik zeer nadrukkelijk bedanken voor de grondige manier waarop zij als reviewers de tekst hebben doorgenomen, gecontroleerd en becommentarieerd. (Dit geldt in het bijzonder voor het nu reeds beruchte Voorbeeld 7.2.) Hun commentaar heeft de leesbaarheid van het boek dan ook duidelijk naar een hoger plan getild.

Suzanne den Ouden wil ik bedanken voor de zeer kundige en keurige wijze waarop zij het typewerk heeft verzorgd. Dat er tussen het aanleveren van de eerste bladzijden en de laatste bladzijden van het manuscript meer dan een jaar verschil zat heeft haar taak er zeker niet

gemakkelijker op gemaakt.

Verder dank ik Jan Willem Nienhuys voor de bereidwilligheid waarmee hij speciaal ten behoeve van dit boek diverse troff-macro's heeft vervaardigd.

Ten slotte wil ik Audrey de Brock-Blank bedanken voor haar morele steun en voor het vele geduld dat zij bij elke verschuiving van de opleveringsdatum van het boek toch telkens weer bleek te bezitten. Ik besef dat zij in de afgelopen perioden de nodige aandacht te kort is gekomen.

Ter afronding van het voorwoord merk ik op dat ik mij uiteraard gaarne aanbevolen houd voor op- en aanmerkingen betreffende het boek.

Waalre, juli 1989

Bert de Brock

0 WISKUNDIGE BASISBEGRIPPEN

In dit hoofdstuk zullen we onze afspraken omtrent de betekenis van de door ons gebruikte basisbegrippen en -notaties vastleggen. We veronderstellen dat voor veel lezers de meeste van deze basisbegrippen wel min of meer bekend zijn; we beperken ons daarom ook meestal slechts tot het geven van de formele definitie van een in te voeren basisbegrip en het opsommen van enige elementaire eigenschappen ervan, gewoonlijk in de vorm van een lemma (zonder bewijs). Wel is het mogelijk dat sommige "bekende" begrippen iets anders (of iets algemener) zijn gedefinieerd dan de lezer gewend is.

Gezien het elementaire karakter van dit hoofdstuk hebben we hier geen opgaven opgenomen. (Wel kan uiteraard elk lemma worden opgevat als een opgave om dat lemma te bewijzen.) De lezer die behoefte heeft aan een uitgebreidere behandeling van de in dit hoofdstuk ingevoerde begrippen, verwijzen we naar bijvoorbeeld [DDS 75]. Ook Deel F1 ("Wiskunde") van [BPZ 87] geeft van deze begrippen diverse eigenschappen en voorbeelden.

0.1 VERZAMELINGEN

Het elementaire begrip *verzameling* veronderstellen we bekend. We memoreren dat een verzameling geen ordening en geen duplicaten kent; ook doet de beschrijvingswijze van de elementen niet ter zake. Er geldt dus bijvoorbeeld

$\{3, 2, 2, 1, 2+1, 2-1\} = \{3, 2, 1\} = \{1, 2, 3\}$.

Ook de betekenis van de volgende uitdrukkingen veronderstellen we bekend:

Uitdrukking	Betekenis
$x \in A$	x is een element van A
$x \notin A$	x is geen element van A
$\mid A \mid$	het aantal elementen van A
$\varnothing$	de lege verzameling
$\mathbb{Z}$	de verzameling van alle gehele getallen
$m \bmod n$	de rest van m na (gehele) deling door n
$m \operatorname{div} n$	m gedeeld door n, met weglating van de rest

Zo geldt voor elke $m \in \mathbb{Z}$ en elke $n \in \mathbb{Z}$ dat
$(m \operatorname{div} n) \in \mathbb{Z}$,
$0 \leq m \bmod n < n$ en
$m = n * (m \operatorname{div} n) + m \bmod n$.
Verder zullen we nog gebruik maken van de volgende afkortingen:

Notatie	Betekenis
$\forall x \in A$:	voor elk element x van A geldt:
$\exists x \in A$:	er is een element x van A waarvoor geldt:
$\Leftrightarrow$	dan en slechts dan als
$\overset{D}{\Leftrightarrow}$	per definitie dan en slechts dan als
$\overset{D}{=}$	is per definitie

In plaats van "$\Leftrightarrow$" schrijven we ook wel eens "desda" (voor "dan en slechts dan als"). Verder merken we op dat we het symbool "$\square$" gebruiken om het einde aan te geven van een voorbeeld, een bewijs of een serie opgaven.

Met bovenstaande notaties kunnen we nu bijvoorbeeld de (ware) bewering dat er voor elk geheel getal een kleiner geheel getal bestaat als volgt op compacte wijze opschrijven:

$$\forall x \in \mathbb{Z} : \exists y \in \mathbb{Z} : y < x$$

Tenslotte veronderstellen we ook het begrip *geordend paar* bekend. Het geordende paar met x als *eerste coördinaat* en y als *tweede coördinaat* noteren we als volgt: $(x\,;y)$. Omgekeerd, als p een gegeven geordend paar is, dan kunnen we de eerste coördinaat van p aanduiden met $\pi_1(p)$ en de tweede coördinaat van p met $\pi_2(p)$.

We memoreren dat geordende paren gelijk zijn dan en slechts dan als zowel hun eerste coördinaten als hun tweede coördinaten gelijk zijn:

$$(x\,;y)=(x'\,;y') \text{ desda } x=x' \text{ en } y=y'.$$

Dus $(2;3)\neq(3;2)$, terwijl $\{2,3\}=\{3,2\}$.

De verzameling van alle natuurlijke getallen (inclusief 0) duiden we aan met $\mathbb{N}$:

DEFINITIE 0.0:
$\mathbb{N} \overset{D}{=} \{k \in \mathbb{Z} \mid k \geq 0\}$.

Als $n \in \mathbb{N}$, dan verstaan we onder Chs(n) de verzameling van alle tekenrijen (ofwel strings) die uit maximaal n tekens bestaan. Voor elke $n \in \mathbb{N}$ is de lege tekenrij, die uit 0 tekens bestaat, een element van Chs(n). Na Voorbeeld 0.2 (aan het eind van dit hoofdstuk) komen we nog op dit begrip terug. Voor elke $n \in \mathbb{N}$ is Chs(n) een eindige verzameling.

We vervolgen onze notatie-afspraken met de formele definities van *verzamelingsinclusie*, de *vereniging*, de *doorsnede*, het *verschil* en het *cartesisch produkt* van twee verzamelingen en tenslotte de *machtsverzameling* van een verzameling:

DEFINITIE 0.1:
Als A en B verzamelingen zijn, dan:

(a) $A \subseteq B \overset{D}{\Longleftrightarrow} \forall x \in A : x \in B$;
(b) $B \supseteq A \overset{D}{\Longleftrightarrow} A \subseteq B$;
(c) $A \subset B \overset{D}{\Longleftrightarrow} A \subseteq B$ en $A \neq B$;
(d) $A \cup B \overset{D}{=} \{x \mid x \in A \text{ of } x \in B\}$;
(e) $A \cap B \overset{D}{=} \{x \mid x \in A \text{ en } x \in B\}$;
(f) $A - B \overset{D}{=} \{x \mid x \in A \text{ en } x \notin B\}$;
(g) $A \times B \overset{D}{=} \{(x\,;y) \mid x \in A \text{ en } y \in B\}$;
(h) $\mathbf{P}(A) \overset{D}{=} \{X \mid X \subseteq A\}$.

We merken op dat de elementen van een verzameling zelf ook verzamelingen kunnen zijn. $\mathbf{P}(A)$, de machtsverzameling van A, is een voorbeeld van zo'n verzameling van verzamelingen.

In de volgende definities introduceren we notaties voor enige nuttige deelverzamelingen van $\mathbb{Z}$:

DEFINITIE 0.2:
Als $m \in \mathbb{Z}$ en $n \in \mathbb{Z}$, dan:

(a) $[m\,..\,n] \stackrel{D}{=} \{k \in \mathbb{Z} \mid m \leq k \text{ en } k \leq n\}$;

(b) $[m\,..\,n) \stackrel{D}{=} \{k \in \mathbb{Z} \mid m \leq k \text{ en } k < n\}$;

(c) $[m\,..) \stackrel{D}{=} \{k \in \mathbb{Z} \mid m \leq k\}$.

DEFINITIE 0.3:
Als $n \in \mathbb{N} - \{0\}$, dan:

(a) $\mathrm{Vng}(n) \stackrel{D}{=} [10^{n-1}\,..\,10^n - 1]$;

(b) $\mathrm{Int}(n) \stackrel{D}{=} [-(10^n - 1)\,..\,10^n - 1]$.

Dus, Vng(n) is de verzameling van alle *natuurlijke* getallen die (in het tientallig stelsel) uit *precies n* cijfers bestaan, en Int(n) is de verzameling van alle *gehele* getallen (ofwel *integers*) die in het tientallig stelsel uit *maximaal n* cijfers bestaan (het teken dus niet meegerekend). Bijvoorbeeld,

Vng(4) = [1000 .. 9999] en Int(4) = [–9999 .. 9999].

Een veel gebruikte manier om sommaties op te schrijven is als volgt:

$$\sum_{\phi_x} u_x$$

waarbij ϕ_x uitdrukt *waarover* er wordt gesommeerd en u_x uitdrukt *wat* er wordt gesommeerd. Omdat in onze toepassingen de uitdrukkingen ϕ_x vaak lang zijn, zullen wij zo'n sommatie aldus schrijven:

$$\sum \phi_x : u_x$$

Zo representeert bijvoorbeeld

$$\sum n \in [0\,..\,199] \text{ en } (n \bmod 3) \neq 0 : n^2$$

de som van de kwadraten van alle natuurlijke getallen kleiner dan 200 die niet deelbaar zijn door drie.

De definitie van de (gegeneraliseerde) vereniging van *een verzameling van* verzamelingen is een generalisatie van de definitie van de vereniging van *twee* verzamelingen (zoals zal blijken in Lemma 0.1 (2)).

DEFINITIE 0.4:
Als W een verzameling van verzamelingen is, dan:
$\bigcup W \stackrel{D}{=} \{x \mid \exists A \in W : x \in A\}$.

Het volgende lemma beschrijft de speciale gevallen waarin de verzameling W uit 0, 1 of 2 elementen bestaat.

LEMMA 0.1:
Als A en B verzamelingen zijn, dan:
(0) $\bigcup \varnothing = \varnothing$;
(1) $\bigcup \{A\} = A$;
(2) $\bigcup \{A, B\} = A \cup B$.

Als R een verzameling geordende paren is, dan verstaan we onder dom(R), *het domein van R*, de verzameling van alle eerste coördinaten in R en onder rng(R), *het bereik van R* of *de range van R*, de verzameling van alle tweede coördinaten in R. Onder R^{-1}, *de inverse van R*, verstaan we de verzameling bestaande uit het "omgekeerde" van elk der elementen van R:

DEFINITIE 0.5:
Als R een verzameling geordende paren is, dan:
(a) $\mathrm{dom}(R) \stackrel{D}{=} \{x \mid (x ; y) \in R\}$;
(b) $\mathrm{rng}(R) \stackrel{D}{=} \{y \mid (x ; y) \in R\}$;
(c) $R^{-1} \stackrel{D}{=} \{(y ; x) \mid (x ; y) \in R\}$.

Een verzameling geordende paren wordt in de wiskunde wel een *relatie* genoemd. We merken hierbij echter meteen op dat dezelfde term in database-theorie vaak wordt gebruikt voor datgene wat wij een *tabel* zullen noemen (in Definitie 1.1).

Het volgende lemma geeft een aantal elementaire eigenschappen van het domein, het bereik en de inverse van enige samengestelde verzamelingen geordende paren weer.

LEMMA 0.2:
Als R en P verzamelingen geordende paren zijn en $D \subseteq R$, dan:

(a) $R \cup P$, $R \cap P$, $R - P$, D en R^{-1} zijn ook verzamelingen geordende paren;

(b) $\text{dom}(R \cup P) = \text{dom}(R) \cup \text{dom}(P)$ en $\text{rng}(R \cup P) = \text{rng}(R) \cup \text{rng}(P)$;

(c) $\text{dom}(R \cap P) \subseteq \text{dom}(R) \cap \text{dom}(P)$ en $\text{rng}(R \cap P) \subseteq \text{rng}(R) \cap \text{rng}(P)$;

(d) $\text{dom}\ (R - P) \supseteq \text{dom}(R) - \text{dom}(P)$ en $\text{rng}(R - P) \supseteq \text{rng}(R) - \text{rng}(P)$;

(e) $\text{dom}(D) \subseteq \text{dom}(R)$ en $\text{rng}(D) \subseteq \text{rng}(R)$;

(f) $\text{dom}(R^{-1}) = \text{rng}(R)$ en $\text{rng}(R^{-1}) = \text{dom}(R)$;

(g) $(R \cup P)^{-1} = R^{-1} \cup P^{-1}$ en $(R \cap P)^{-1} = R^{-1} \cap P^{-1}$ en
$(R - P)^{-1} = R^{-1} - P^{-1}$ en $(R^{-1})^{-1} = R$.

0.2 FUNCTIES

We definiëren het bekende begrip *functie* op de volgende wijze:

DEFINITIE 0.6:
F is een **functie** $\overset{D}{\Longleftrightarrow}$ F is een verzameling geordende paren en
$$\forall(x\,;y) \in F : \forall(x'\,;y') \in F : \text{ als } x = x' \text{ dan } y = y'.$$

Met andere woorden, F is een functie als F een verzameling geordende paren is waarvoor er bij elke $x \in \text{dom}(F)$ *precies één* y bestaat met $(x\,;y) \in F$. Voor elke functie f en elke $x \in \text{dom}(f)$ is $f(x)$ per definitie de eenduidig bepaalde y waarvoor geldt dat $(x\,;y) \in f$. Lees $f(x)$ als: *f toegepast op x*. Omgekeerd is f inv y, *het origineel van y onder f*, de verzameling van alle $x \in \text{dom}(f)$ waarvoor geldt dat $(x\,;y) \in f$. Dus, $f \text{ inv } y = \{x \in \text{dom}(f) \mid f(x) = y\}$.

Een element van $\text{dom}(f)$ noemen we wel een *argument van f* en een element van $\text{rng}(f)$ wel een *waarde van f*. In het bijzonder noemen we $f(x)$ wel *de waarde van f in x*. Als $f(x) = x$ dan noemen we x wel een *dekpunt van f*.

Ter illustratie definiëren we de verzameling f1 als volgt:

f1 = {(EMPNO; 10), (DEPNO; 6), (SAL; 1962)}

Volgens Definitie 0.6 is f1 een functie en f1(EMPNO) = 10, f1(DEPNO) = 6 en f1(SAL) = 1962. De verzameling {(EMPNO; 10), (DEPNO; 6), (SAL; 1962), (DEPNO; 8)} is geen functie. Een eenvoudig (en klassiek) voorbeeld van een functie is $\{(x\,;x+1) \mid x \in \mathbb{N}\}$, de zogeheten *successorfunctie* voor de natuurlijke getallen. De verzameling $\{((k\,;n)\,;k^2+n) \mid (k\,;n) \in \mathbb{Z} \times \mathbb{N}\}$ is een voorbeeld van een functie "in twee variabelen".

Omdat elke functie een verzameling geordende paren is, is elk begrip en elk lemma dat van toepassing is op verzamelingen in het algemeen of op verzamelingen geordende paren in het bijzonder, ook van toepassing op functies! Zo zijn bijvoorbeeld de doorsnede en de vereniging van twee functies gedefinieerd, evenals het domein en het bereik van een functie. Bijvoorbeeld, dom(f1) = {EMPNO, DEPNO, SAL} en rng(f1) = {6, 10, 1962}; van de eerder genoemde functie "in twee variabelen" is $\mathbb{Z} \times \mathbb{N}$ het domein en $\mathbb{N}$ het bereik.

Verder is ook Lemma 0.2 van toepassing op functies. Zo zien we dat voor alle functies f en g zowel $f \cup g$ als $f \cap g$ en $f - g$ verzamelingen geordende paren zijn. Maar zijn het ook *functies*? Op deze vraag gaan we nu nader in.

LEMMA 0.3:
Als f en g functies zijn en $D \subseteq f$, dan:

(a) D is een functie;

(b) $f \cap g$ en $f - g$ zijn functies;

(c) $\mid \mathrm{rng}(f) \mid \leq \mid \mathrm{dom}(f) \mid$;

(d) $\varnothing$ is een functie en $\mathrm{dom}(\varnothing) = \mathrm{rng}(\varnothing) = \varnothing$.

De vereniging van twee functies hoeft blijkbaar geen functie te zijn. Ter illustratie definiëren we de functie f2 als volgt:

f2 = {(DEPNO; 6), (SAL; 2100), (AGE; 26)}.

Nu is f1 $\cup$ f2 geen functie, want zowel (SAL; 1962) $\in$ f1 $\cup$ f2 als (SAL; 2100) $\in$ f1 $\cup$ f2.

Als f en g functies zijn waarvoor wel geldt dat $f \cup g$ een functie is, dan noemen we f en g *compatibel*:

DEFINITIE 0.7:
Als f en g functies zijn, dan:
f **en** g **zijn compatibel** $\overset{D}{\Longleftrightarrow} f \cup g$ is een functie.

In de Engelstalige database-literatuur worden functies die compatibel zijn ook wel *joinable* genoemd. We merken op dat elke functie compatibel met zichzelf is. Als de functies f en g compatibel zijn en de functies g en h ook compatibel zijn, dan hoeven f en h nog niet compatibel te zijn. (Ga dit na.) In Lemma 0.12 zullen we nog enige criteria geven opdat twee functies compatibel zijn.

Het volgende lemma geeft nuttige criteria voor het bewijzen van inclusie of gelijkheid van functies:

LEMMA 0.4:
Als f en g functies zijn, dan:

(a) $f \subseteq g$ desda $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ en $\forall x \in \mathrm{dom}(f) : f(x) = g(x)$;

(b) $f = g$ desda $\mathrm{dom}(f) = \mathrm{dom}(g)$ en $\forall x \in \mathrm{dom}(f) : f(x) = g(x)$.

Onderdeel (a) is af te leiden uit Definitie 0.6 en onderdeel (b) volgt uit (a).

Om het spreken over functies gemakkelijker te maken, introduceren we de volgende nomenclatuur:

DEFINITIE 0.8:
Als A een verzameling is, dan:

(a) f is een **functie over** $A \overset{D}{\Longleftrightarrow} f$ is een functie en $\mathrm{dom}(f) = A$;

(b) f is een **functie uit** $A \overset{D}{\Longleftrightarrow} f$ is een functie en $\mathrm{dom}(f) \subseteq A$;

(c) f is een **functie naar** $A \overset{D}{\Longleftrightarrow} f$ is een functie en $\mathrm{rng}(f) \subseteq A$;

(d) f is een **functie op** $A \overset{D}{\Longleftrightarrow} f$ is een functie en $\mathrm{rng}(f) = A$.

In de literatuur wordt een functie uit A ook wel een *partiële* functie over A genoemd. Een functie over A noemen we ook wel een functie *van* A in uitdrukkingen zoals "functie *van* A *naar* B" en "functie *van* A *op* B".

We merken op dat elke functie f een functie *van* $\mathrm{dom}(f)$ *op* $\mathrm{rng}(f)$ is.

Voor elke verzameling A definiëren we $\mathrm{id}(A)$, *de identieke functie op* A, als de functie die aan elk element van A dat element zelf toevoegt:

DEFINITIE 0.9:
Als A een verzameling is, dan:
$\mathrm{id}(A) \stackrel{D}{=} \{(x\,;x) \mid x \in A\}$.

LEMMA 0.5:
Als A en B verzamelingen zijn, dan:

(a) $\mathrm{id}(A)$ is een functie van A op A;

(b) $\mathrm{id}(A \cup B) = \mathrm{id}(A) \cup \mathrm{id}(B)$;

(c) $\mathrm{id}(A \cap B) = \mathrm{id}(A) \cap \mathrm{id}(B)$;

(d) $\mathrm{id}(A - B) = \mathrm{id}(A) - \mathrm{id}(B)$.

De verzameling van alle functies van A naar B geven we aan met $A \twoheadrightarrow B$ en de verzameling van alle "partiële" functies van A naar B met $A \,\tilde{\twoheadrightarrow}\, B$:

DEFINITIE 0.10:
Als A en B verzamelingen zijn, dan:

(a) $A \twoheadrightarrow B \stackrel{D}{=} \{f \mid f \text{ is een functie en } \mathrm{dom}(f) = A \text{ en } \mathrm{rng}(f) \subseteq B\}$;

(b) $A \,\tilde{\twoheadrightarrow}\, B \stackrel{D}{=} \{f \mid f \text{ is een functie en } \mathrm{dom}(f) \subseteq A \text{ en } \mathrm{rng}(f) \subseteq B\}$.

Uit de definitie volgt dat $(A \twoheadrightarrow B) \subseteq (A \,\tilde{\twoheadrightarrow}\, B)$. We kunnen deze notaties herhaald toepassen; zo is $A \twoheadrightarrow (B \twoheadrightarrow C)$ de verzameling van alle functies die aan elk element van A een functie van B naar C toevoegen. Daarentegen is $(A \twoheadrightarrow B) \twoheadrightarrow C$ de verzameling van alle functies die aan elke functie van A naar B een element van C toevoegen.

In de literatuur wordt "$f \in A \twoheadrightarrow B$" vaak geschreven als "$f : A \twoheadrightarrow B$" en $A \twoheadrightarrow B$ wel als B^A.

Als R een verzameling geordende paren is dan noemen we R *injectief* als er bij elke $y \in \mathrm{rng}(R)$ *precies één* x bestaat met $(x\,;y) \in R$:

DEFINITIE 0.11:
Als R een verzameling geordende paren is, dan:
R **is injectief** $\stackrel{D}{\Longleftrightarrow} \forall (x\,;y) \in R : \forall (x'\,;y') \in R$: als $y = y'$ dan $x = x'$.

Merk op dat bovenstaande eis in feite de omkering is van die in Definitie 0.6.

Voor functies komt injectiviteit op het volgende neer:

LEMMA 0.6:
Als f een functie is, dan:
f is injectief $\Longleftrightarrow \forall x \in \text{dom}(f) : \forall x' \in \text{dom}(f)$: als $f(x) = f(x')$ dan $x = x'$.

Een injectieve functie wordt ook wel een *injectie* of een *één-éénduidige functie* genoemd.

LEMMA 0.7:
Als R een verzameling geordende paren is, dan:

(a) R is injectief desda R^{-1} is een functie;

(b) als R een injectieve functie is, dan $|\text{dom}(R)| = |\text{rng}(R)|$.

Naar aanleiding van onderdeel (b) van het voorgaande lemma (en onderdeel (c) van Lemma 0.3) memoreren we nog de volgende nuttige equivalenties:

LEMMA 0.8:
Als A en B verzamelingen zijn, dan:

(a) $|A| \geq |B|$ desda $\exists f \in A \twoheadrightarrow B : \text{rng}(f) = B$;

(b) $|A| \leq |B|$ desda $\exists f \in A \twoheadrightarrow B : f$ is injectief;

(c) $|A| = |B|$ desda $\exists f \in A \twoheadrightarrow B : f$ is injectief en $\text{rng}(f) = B$.

De onder (c) genoemde functie wordt wel een *bijectie van A op B* genoemd:

DEFINITIE 0.12:
Als A en B verzamelingen zijn, dan:
f is een **bijectie van** A **op** $B \overset{D}{\Longleftrightarrow} f$ is een injectieve functie en
$\text{dom}(f) = A$ en $\text{rng}(f) = B$.

We zullen de notatie "$\lambda x \in A : u_x$" (waarbij u_x een of andere uitdrukking in x voorstelt) soms gebruiken als afkorting voor "$\{(x ; u_x) \mid x \in A\}$". Dus $\lambda x \in A : u_x$ representeert een functie over A. Zo kunnen we bijvoorbeeld de successorfunctie voor de natuurlijke getallen schrijven als $\lambda x \in \mathbb{N} : x + 1$ en de eerder genoemde functie "in twee variabelen" als $\lambda (k ; n) \in \mathbb{Z} \times \mathbb{N} : k^2 + n$. Verder kunnen we voor elke verzameling A de identieke functie op A nu ook als volgt omschrijven:

$$\text{id}(A) = \lambda x \in A : x$$

Verder gebruiken we ook nog de notatie "$\lambda x \in A$ en $\phi_x : u_x$" (waarbij ϕ_x een of andere voorwaarde voor x voorstelt) als afkorting voor "$\{(x ; u_x) \mid x \in A \text{ en } \phi_x\}$".

We definiëren *de samenstelling* (of *compositie*) *van g na f* voor elk tweetal functies f en g (en niet alleen maar indien $\mathrm{rng}(f) \subseteq \mathrm{dom}(g)$, zoals gebruikelijk is):

DEFINITIE 0.13:
Als f en g functies zijn, dan:
$g \circ f \overset{D}{=} \{(x\,;\, g(f(x))) \mid x \in \mathrm{dom}(f) \text{ en } f(x) \in \mathrm{dom}(g)\}$.

In λ-notatie: $g \circ f = \lambda x \in \mathrm{dom}(f) \text{ en } f(x) \in \mathrm{dom}(g) : g(f(x))$.

Een van de toepassingen van deze (wat ruimere) definitie van functiecompositie illustreren we aan de hand van de onderstaande functies h1 en (de reeds eerder genoemde) f1:

h1 = {(AFDNR; DEPNO), (MEDEWNR; EMPNO), (WPL; CITY)}
f1 = {(DEPNO; 6), (EMPNO; 10), (SAL; 1962)}

Uit bovenstaande definitie van functiecompositie volgt nu dat f1 $\circ$ h1 de verzameling {(AFDNR; 6), (MEDEWNR; 10)} is. Verder geldt h1 $\circ$ f1 = $\varnothing$ want er is geen enkele $x \in$ dom(f1) met f1(x) $\in$ dom(h1).

Van het volgende lemma, dat diverse nuttige eigenschappen van functiecompositie opsomt, zal in latere hoofdstukken herhaaldelijk gebruik worden gemaakt.

LEMMA 0.9:
Als f, g en h functies zijn, dan:
(a) $g \circ f$ is een functie;
(b) $\mathrm{dom}(g \circ f) \subseteq \mathrm{dom}(f)$;
(c) $\mathrm{rng}(g \circ f) \subseteq \mathrm{rng}(g)$;
(d) als $\mathrm{rng}(f) \subseteq \mathrm{dom}(g)$, dan $\mathrm{dom}(g \circ f) = \mathrm{dom}(f)$;
(e) als $\mathrm{dom}(g) \subseteq \mathrm{rng}(f)$, dan $\mathrm{rng}(g \circ f) = \mathrm{rng}(g)$;
(f) $h \circ (g \circ f) = (h \circ g) \circ f$;
(g) als $\mathrm{dom}(f) = \mathrm{dom}(g)$, dan $\mathrm{dom}(f \circ h) = \mathrm{dom}(g \circ h)$;
(h) $f \circ h = g \circ h \Leftrightarrow f \circ \mathrm{id}(\mathrm{rng}(h)) = g \circ \mathrm{id}(\mathrm{rng}(h))$;
(i) als $f \circ h = g \circ h$ en $\mathrm{dom}(f) \subseteq \mathrm{rng}(h)$, dan $f \subseteq g$;
(j) als $f \circ h = g \circ h$ en $\mathrm{dom}(f) = \mathrm{dom}(g) \subseteq \mathrm{rng}(h)$, dan $f = g$.

Uit het volgende lemma blijkt dat compositie met de identieke functie op B, zowel "vooraf"

als "achteraf", als het ware als een "filter" werkt. Verder zien we dat de compositie van de identieke functies op twee verzamelingen de identieke functie op de doorsnede van die verzamelingen is.

LEMMA 0.10:
Als f een functie is en A en B zijn verzamelingen, dan:

(a) $f \circ \mathrm{id}(B) = \{(x\,;y) \in f \mid x \in B\}$;

(b) $\mathrm{id}(B) \circ f = \{(x\,;y) \in f \mid y \in B\}$;

(c) $\mathrm{id}(A) \circ \mathrm{id}(B) = \mathrm{id}(A \cap B)$.

Voor elke functie f en elke verzameling B noemen we de zojuist onder (a) genoemde verzameling *de restrictie van f tot B*, met andere woorden de verzameling van alle elementen van f waarvan de eerste coördinaat in B ligt. De restrictie van f tot B duiden we aan met $f \upharpoonright B$; het andere deel van f duiden we aan met $f \not\upharpoonright B$:

DEFINITIE 0.14:
Als f een functie is en B is een verzameling, dan:

(a) $f \upharpoonright B \overset{D}{=} \{(x\,;y) \in f \mid x \in B\}$;

(b) $f \not\upharpoonright B \overset{D}{=} \{(x\,;y) \in f \mid x \notin B\}$.

Bijvoorbeeld, f1 $\upharpoonright$ {EMPNO, SAL} = {(EMPNO; 10), (SAL; 1962)} en f1 $\not\upharpoonright$ {EMPNO, SAL} = {(DEPNO; 6)}; voor de functie h1 geldt dat h1 $\upharpoonright$ {EMPNO, SAL} = ∅ en h1 $\not\upharpoonright$ {EMPNO, SAL} = h1 (omdat EMPNO en SAL in h1 niet als eerste coördinaten voorkomen).

LEMMA 0.11:
Als f een functie is en B en B' zijn verzamelingen, dan:

(a) $f \upharpoonright B$ is een functie over $\mathrm{dom}(f) \cap B$;

(b) $f \not\upharpoonright B$ is een functie over $\mathrm{dom}(f) - B$;

(c) $f \upharpoonright B$ en $f \not\upharpoonright B$ zijn compatibel;

(d) $f = (f \upharpoonright B) \cup (f \not\upharpoonright B)$;

(e) $f \upharpoonright B = \lambda x \in \mathrm{dom}(f) \cap B : f(x)$;

(f) $f \not\upharpoonright B = \lambda x \in \mathrm{dom}(f) - B : f(x)$;

(g) $f \not\upharpoonright B = f \upharpoonright (\mathrm{dom}(f) - B)$;

(h) $f \upharpoonright B = f \circ \mathrm{id}(B)$;

(i) $(f \not\upharpoonright B) \not\upharpoonright B' = f \not\upharpoonright (B \cup B')$;

(j) $(f \upharpoonright B) \upharpoonright B' = f \upharpoonright (B \cap B')$;

(k) $(f \upharpoonright B) \not\upharpoonright B' = f \upharpoonright (B - B')$;

(l) $f \upharpoonright \varnothing = \varnothing$ en $\varnothing \upharpoonright B = \varnothing$;

(m) $f \not\upharpoonright \varnothing = f$ en $\varnothing \not\upharpoonright B = \varnothing$.

Lemma 0.11(h) vertelt nog eens expliciet dat restrictie eigenlijk een speciaal geval van functiecompositie is.

DEFINITIE 0.15:
Als f en g functies zijn en A is een verzameling, dan:
$f \equiv g$ **op** $A \overset{D}{\Longleftrightarrow} f \upharpoonright A = g \upharpoonright A$.

We zeggen in bovenstaand geval wel dat f *gelijk is aan* g *op* A of dat f en g *overeenstemmen op* A.

Soms is een gegeven functie f "niet helemaal" de functie die we willen hebben, in die zin dat er sommige paren $(x\,;y)$ ontbreken of dat er bij sommige argumenten nog niet de "goede" waarden horen. De op f toe te passen "correctie" kunnen we representeren door een functie. Het "correctieresultaat" definiëren we als volgt:

DEFINITIE 0.16:
Als f en g functies zijn, dan:
$f \,\theta\, g \overset{D}{=} (f \not\upharpoonright \mathrm{dom}(g)) \cup g$.

Bijvoorbeeld, voor de eerder genoemde functies f1 en f2 geldt het volgende:

f1 θ f2 = {(EMPNO; 10), (DEPNO; 6), (SAL; 2100), (AGE; 26)} en
f2 θ f1 = {(EMPNO; 10), (DEPNO; 6), (SAL; 1962), (AGE; 26)}

We noemen $f \,\theta\, g$ wel *de mutatie* (of *modificatie) van* f *volgens* g.

LEMMA 0.12:
Als f, g en h functies zijn, dan:

(a) $f \theta g$ is een functie over $\mathrm{dom}(f) \cup \mathrm{dom}(g)$;

(b) als $\mathrm{dom}(f) = \mathrm{dom}(g)$, dan $f \theta g = g$;

(c) $(f \theta g) \theta h = f \theta (g \theta h)$;

(d) $f \theta \varnothing = \varnothing \theta f = f$.

We zijn nu ook in staat enige alternatieve omschrijvingen van compatibiliteit te geven.

LEMMA 0.13:
Als f en g functies zijn, dan zijn de volgende beweringen equivalent:

(a) f en g zijn compatibel;

(b) $f \cup g$ is een functie;

(c) f en g stemmen overeen op $\mathrm{dom}(f) \cap \mathrm{dom}(g)$;

(d) $\mathrm{dom}(f \cap g) = \mathrm{dom}(f) \cap \mathrm{dom}(g)$;

(e) $f \theta g = g \theta f$;

(f) $f \theta g = f \cup g$.

Vanaf onderdeel (c) geeft dit lemma dus als het ware vier alternatieve definities van compatibiliteit. Uit het lemma blijkt ook dat compatibele functieparen diverse interessante eigenschappen hebben. In Hoofdstuk 2 komen we op compatibele functieparen nog uitgebreid terug. Het volgende lemma geeft nog een aantal eigenschappen van compatibele functieparen weer.

LEMMA 0.14:
Als f, g en h functies zijn en f en g zijn compatibel, dan:

(a) $f \circ h$ en $g \circ h$ zijn compatibel;

(b) $(f \circ h) \cup (g \circ h) = (f \cup g) \circ h$;

(c) $(h \circ f) \cup (h \circ g) = h \circ (f \cup g)$.

We merken op dat de waarde van een functie een verzameling of zelf(s) een functie kan zijn. Bijvoorbeeld, $\lambda n \in \mathbb{N}$: $[0 \mathinner{..} (n-1)]$ is de "verzamelingswaardige" functie over $\mathbb{N}$ die aan elk natuurlijk getal de verzameling van al zijn voorgangers in $\mathbb{N}$ toevoegt, en $\lambda n \in \mathbb{N} : (\lambda k \in \mathbb{Z} : k^n)$ is de "functiewaardige" functie over $\mathbb{N}$ die aan elk natuurlijk getal n de n-de machts-functie over $\mathbb{Z}$ toevoegt.

Een verzamelingswaardige(*) functie noemen we ook wel gewoon een *verzamelingsfunctie*:

DEFINITIE 0.17:
F is een **verzamelingsfunctie** $\overset{D}{\Longleftrightarrow}$ F is een functie en
$\forall x \in \mathrm{dom}(F) : F(x)$ is een verzameling.

VOORBEELD 0.1:

Ter illustratie geven we twee voorbeelden van verzamelingsfuncties. Deze functies, FM en FA, zullen we ook later nog herhaaldelijk gebruiken. Om "de fantasie te prikkelen" geven we na de verticale lijn aan waaraan u zou kunnen denken bij dit voorbeeld.

FM = {(NR ; $\mathbb{N}$),	identiteitsnummer van een *medewerker*
(NAAM ; Chs(40)),	naam
(SAL ; $\mathbb{N}$),	salaris
(GESL ; { ‘M’ , ‘V’ }),	geslacht
(AFDNR ; [1 .. 99])}	afdelingsnummer
FA = { (ANR ; $\mathbb{N}$),	nummer van een *afdeling*
(NAAM ; Chs(45)),	afdelingsnaam
(MANNR ; $\mathbb{N}$)}	id. nummer van de manager

□ Voorbeeld 0.1.

Als F een verzamelingsfunctie is, dan verstaan we onder $\Pi(F)$ de verzameling van alle functies over dom(F) die voor elke $x \in \mathrm{dom}(F)$ een "keuze" uit de verzameling $F(x)$ maken, en onder $\bar{\Pi}(F)$ de verzameling van alle functies die voor "sommige" $x \in \mathrm{dom}(F)$ een keuze uit $F(x)$ maken:

(*) Voor lezers die bekend zijn met de axiomatische verzamelingenleer merken we op dat we hier een naïeve verzamelingenleer gebruiken waarin we niet vooronderstellen dat alles een verzameling is.

DEFINITIE 0.18:
Als F een verzamelingsfunctie is, dan:

(a) $\Pi(F) \stackrel{D}{=} \{f \mid f$ is een functie over $\mathrm{dom}(F)$ en $\forall x \in \mathrm{dom}(f) : f(x) \in F(x)\}$;

(b) $\tilde{\Pi}(F) \stackrel{D}{=} \{f \mid f$ is een functie uit $\mathrm{dom}(F)$ en $\forall x \in \mathrm{dom}(f) : f(x) \in F(x)\}$.

We noemen $\Pi(F)$ wel *het gegeneraliseerde produkt van F*. Uit de definitie volgt dat $\Pi(F) \subseteq \tilde{\Pi}(F)$.

Voor de eerder genoemde verzamelingsfunctie FA is Π(FA) de verzameling van alle functies van de vorm

$\{$(ANR; n), (NAAM; α), (MANNR; m)$\}$,

waarbij $n \in \mathbb{N}$, $\alpha \in$ Chs(45) en $m \in \mathbb{N}$.

Voor alle verzamelingen A en B kunnen we met behulp van Definitie 0.18 en de functie $\lambda a \in A : B$, dus de (constante) verzamelingsfunctie die aan elk element van A de verzameling B toevoegt, de in Definitie 0.10 ingevoerde begrippen ook als volgt beschrijven:

LEMMA 0.15:
Als A en B verzamelingen zijn, dan:

(a) $A \twoheadrightarrow B = \Pi(\lambda a \in A : B)$;

(b) $A \tilde{\twoheadrightarrow} B = \tilde{\Pi}(\lambda a \in A : B)$.

Als V een willekeurige verzameling functies is dan verstaan we onder He(V), *de heading van V* (of *het totale domein van V*), de vereniging van de domeinen van elk der elementen van V:

DEFINITIE 0.19:
Als V een verzameling functies is, dan:
$\mathrm{He}(V) \stackrel{D}{=} \bigcup \{\mathrm{dom}(f) \mid f \in V\}$.

Zo geldt bijvoorbeeld dat He({f1, f2}) = {EMPNO, DEPNO, SAL, AGE}, waarbij

f1 = {(EMPNO; 10), (DEPNO; 6), (SAL; 1962)}
f2 = {(DEPNO; 6), (SAL; 2100), (AGE; 26)}

Het volgende lemma behandelt enige speciale gevallen.

LEMMA 0.16:
Als A en B verzamelingen zijn en F is een verzamelingsfunctie, dan:

(a) He $(\Pi(F))$ $= \mathrm{dom}(F)$ als $\Pi(F) \neq \varnothing$;

(b) He $(A \mathrel{-\!\!\triangleright} B)$ $= A$ als $B \neq \varnothing$;

(c) He $(A \mathrel{-\!\!\triangleright} \varnothing)$ $= \varnothing$;

(d) He $(\varnothing)$ $= \varnothing$.

0.3 DE TRANSITIEVE AFSLUITING VAN EEN RELATIE

Alvorens we de transitieve afsluiting van een relatie definiëren voeren we eerst het begrip *rij* in:

DEFINITIE 0.20:
Als $n \in \mathbb{N}$, dan:
r is een **rij ter lengte** n $\overset{D}{\Longleftrightarrow}$ r is een functie en $\mathrm{dom}(r) = [0\,..\,n)$.

VOORBEELD 0.2:

Voorbeelden van rijen zijn:

r1 = {(0; 37), (1; 29), (2; 37)}
r2 = {(0; 29), (1; 37), (2; 37)}
r3 = {(0; 29), (1; 37), (2; 37), (3; 29)}

De functies r1 en r2 zijn rijen ter lengte 3 en r3 is een rij ter lengte 4.
We noteren rijen soms ook wel als volgt:

r1 = <37; 29; 37>
r2 = <29; 37; 37>
r3 = <29; 37; 37; 29>

We zien dat r1 ≠ r2, r2 ≠ r3 en r3 ≠ r1.

☐ Voorbeeld 0.2.

Voor elke $n \in \mathbb{N}$ vatten we ook Chs(n) op als een verzameling rijen:

$$\mathrm{Chs}(n) = \{r \mid \exists k \in [0 \,..\, n]: r \text{ is een rij ter lengte } k \text{ en } \mathrm{rng}(r) \subseteq C\},$$

waarbij C de (niet nader gespecificeerde) verzameling tekens is. Als we ter illustratie in Voorbeeld 0.2 het getal 37 door het 'E'-teken vervangen en het getal 29 door het 'N'-teken, dan krijgen we de representaties van de strings 'ENE', 'NEE' en 'NEEN', respectievelijk.

We merken op dat rijen kunnen worden gebruikt wanneer de eventuele duplicaten en de ordening der objecten relevant zijn, zoals in de voorgaande voorbeelden van getallenrijen en tekenrijen (waarin immers r1 $\neq$ r2 en r2 $\neq$ r3).

Als r en r' rijen zijn met $r \subseteq r'$, dan noemen we r een *beginstuk van* r' en, omgekeerd, r' een *verlenging van* r. De naamgeving moge duidelijk worden uit Voorbeeld 0.2, waarin r2 $\subseteq$ r3. Ook als r en r' elementen van Chs(n) zijn, dan drukt $r \subseteq r'$ inderdaad uit dat de tekenrij r een beginstuk van de tekenrij r' vormt; zie het zojuist gegeven voorbeeld van 'NEE' en 'NEEN'.

We beëindigen de behandeling van rijen met de opmerking dat uit Definitie 0.20 blijkt dat de lengte van een rij r altijd $|r|$ is en dat $\varnothing$ de enige rij ter lengte 0 is.

DEFINITIE 0.21:
Als $n \in \mathbb{N} - \{0\}$ en R is een relatie, dan:
$R \uparrow n \overset{D}{=} \{(r(0); r(n)) \mid r \text{ is een rij ter lengte } n+1 \text{ en } \forall i \in [0 \,..\, n): (r(i); r(i+1)) \in R\}$.

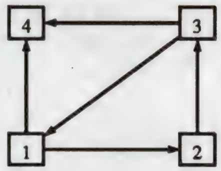
LEMMA 0.17:
(a) $R \uparrow 1 = R$ voor elke relatie R;
(b) $\varnothing \uparrow n = \varnothing$ voor elke $n \in \mathbb{N} - \{0\}$.

VOORBEELD 0.3:

We definiëren de relatie R1 als volgt:

R1 = {(1; 2), (2; 3), (3; 4), (1; 4), (3; 1)}

We kunnen zo'n verzameling geordende paren ook grafisch weergeven:



Figuur 0.1: Een grafische weergave van R1

Uit Definitie 0.21 leiden we af dat

R1 $\uparrow$ 2 = {(1; 3), (2; 4), (2; 1), (3; 2), (3; 4)},
R1 $\uparrow$ 3 = {(1; 4), (1; 1), (2; 2), (2; 4), (3; 3)} en
R1 $\uparrow$ 4 = R1

Verder geldt voor elke $m \in \mathbb{N}$ en elke $k \in \{1,2,3\}$ dat R1 $\uparrow (3m+k) =$ R1 $\uparrow k$.

□ Voorbeeld 0.3.

In termen van de grafische weergave van R kunnen we $R \uparrow n$ nu informeel omschrijven als de verzameling (beginpunt; eindpunt)-paren van alle "wandelingen" van precies n stappen die in de grafische weergave van R mogelijk zijn.

Onder *de transitieve afsluiting van* een relatie R verstaan we de verzameling (beginpunt; eindpunt)-paren van alle mogelijke *niet-lege* wandelingen in de grafische weergave van R. Voor de transitieve afsluiting van R gebruiken we de notatie Tcl(R), van het Engelse "transitive closure". We noemen R *cyclisch* als er echte "rondwandelingen" in R mogelijk zijn; in het andere geval noemen we R *acyclisch.*

DEFINITIE 0.22:
Als R een relatie is, dan:
(a) $\mathrm{Tcl}(R) \overset{D}{=} \bigcup \{R \uparrow n \mid n \in \mathbb{N} - \{0\}\}$;
(b) R **is cyclisch** $\overset{D}{\Longleftrightarrow} \exists (x;y) \in \mathrm{Tcl}(R) : x = y$;
(c) R **is acyclisch** $\overset{D}{\Longleftrightarrow} \forall (x;y) \in \mathrm{Tcl}(R) : x \neq y$.

Wanneer we Definitie 0.22 (a) uitschrijven dan zien we (met behulp van Definitie 0.21) dat $(x;y) \in \mathrm{Tcl}(R)$ indien er een n in $\mathbb{N} - \{0\}$ en een rij r ter lengte $n+1$ zijn zodanig dat $x = r(0)$ en $y = r(n)$ en $(r(i); r(i+1)) \in R$ voor alle i in $[0 \,..\, n)$.

Voor R1 uit Voorbeeld 0.3 geldt dat

$\mathrm{Tcl}(\mathrm{R1}) = (\mathrm{R1} \uparrow 1) \cup (\mathrm{R1} \uparrow 2) \cup (\mathrm{R1} \uparrow 3) = \{1, 2, 3\} \times \{1, 2, 3, 4\}$.

De relatie R1 is dus cyclisch.

LEMMA 0.18:
Als R een relatie is, dan:

(a) $R \subseteq \mathrm{Tcl}(R)$;

(b) $\mathrm{Tcl}(\mathrm{Tcl}(R)) = \mathrm{Tcl}(R)$;

(c) $\mathrm{Tcl}(\varnothing) = \varnothing$;

(d) als R acyclisch is, dan is R *irreflexief* (d.w.z. $\forall (x\,; y) \in R : x \neq y$);

(e) R is acyclisch $\Longleftrightarrow$ $\mathrm{Tcl}(R)$ is irreflexief.

1 VAN TABELLEN TOT DATABASE-UNIVERSA

In dit hoofdstuk zullen we achtereenvolgens de belangrijke basisbegrippen *tabel*, *database-toestand* en *database-universum* introduceren.

1.1 TABELLEN

Onder een *tabel over* een verzameling A verstaan we een verzameling functies over A:

DEFINITIE 1.1:
Als A een verzameling is, dan:
T is een **tabel over** $A \overset{D}{\Longleftrightarrow} T$ is een verzameling en
$\forall t \in T : t$ is een functie over A.

Een element van een tabel wordt ook wel een *tupel* genoemd.

VOORBEELD 1.1:

Figuur 1.1(a) representeert een tabel T1 over {AFDNR, GESL, NAAM, NR, SAL} en Figuur 1.1(b) een tabel T2 over {ANR, MANNR, NAAM}. T1 heeft drie elementen en T2 heeft twee elementen.

NR	NAAM	SAL	GESL	AFDNR
8	JANSSEN	2200	M	1
7	JANSSEN	3109	V	2
9	DEKKER	2995	M	1

(a) Een tabel over {NR, NAAM, SAL, GESL, AFDNR}

ANR	NAAM	MANNR
2	PLANNING	7
1	PRODUCTIE	9

(b) Een tabel over {ANR, NAAM, MANNR}

Figuur 1.1

De twee elementen van T2 zijn de functies

t1 = {(ANR; 2), (MANNR; 7), (NAAM; 'PLANNING')} en
t2 = {(ANR; 1), (MANNR; 9), (NAAM; 'PRODUCTIE')}.

Dus t1(ANR) = 2, t1(MANNR) = 7, etcetera. We zien dat dom(t1) = dom(t2) = {ANR, MANNR, NAAM}, in overeenstemming met Definitie 1.1.

Merk op dat T1 $\subseteq \Pi$(FM) en T2 $\subseteq \Pi$(FA), waarbij FM en FA de in Voorbeeld 0.1 gedefinieerde verzamelingswaardige functies zijn.

Bij NR, NAAM, SAL, GESL en AFDNR in Figuur 1.1(a) mag men achtereenvolgens denken aan medewerkersnummer, naam, salaris, geslacht en afdelingsnummer van een medewerker. Bij ANR, NAAM en MANNR in Figuur 1.1(b) mag men achtereenvolgens denken aan afdelingsnummer, afdelingsnaam en medewerkersnummer van de manager van de afdeling. Noodzakelijk voor het formele begrip zijn deze "bijgedachten" uiteraard niet.

□ Voorbeeld 1.1.

Omdat elke tabel een (speciale) verzameling is, is elk begrip dat is gedefinieerd voor verzamelingen ook van toepassing op tabellen. Zo kunnen we bijvoorbeeld spreken over de vereniging en de doorsnede van twee tabellen; zie ook § 2.1.1.

Uit Definitie 1.1. volgt dat $\varnothing$ een tabel is over *elke* verzameling. (Ga dit na.) Echter, voor elke verzameling A en elke *niet-lege* tabel T over A is A de enige verzameling waarover T een tabel is. Het bewijs gaat als volgt:
Stel dat T een tabel over A *en* over A' is en dat $T \neq \varnothing$. Dan is er een $t_0 \in T$; nu is t_0 een functie over A en over A'. Dus $\mathrm{dom}(t_0) = A$ en $\mathrm{dom}(t_0) = A'$. Conclusie: $A' = A$. Dus A is de enige verzameling waarover T een tabel is.

De opgaven 1.1 en 1.2 geven alternatieve omschrijvingen voor de begrippen "tabel" en "tabel over A".

OPGAVEN

1.1. Bewijs de onderstaande equivalentie (Hint: neem voor het "$\Leftarrow$"-gedeelte $A = \mathrm{He}(V)$.)

$$\text{V is een tabel} \iff \text{V is een verzameling functies en}$$
$$\forall f \in V : \forall f' \in V : \mathrm{dom}(f) = \mathrm{dom}(f').$$

1.2. Als A een verzameling is, dan:
T is een tabel over A $\iff$ $T \subseteq A \twoheadrightarrow B$ voor een of andere verzameling B.
Bewijs dit. (Hint: neem voor het "$\Rightarrow$"-gedeelte $B = \bigcup \{\mathrm{rng}(t) \mid t \in T\}$.)

□

1.2 DATABASE-TOESTANDEN

VOORBEELD 1.2:

Figuur 1.1 kan worden opgevat als een weergave van (het relevante deel van) de toestand van een zekere (kleine) organisatie. Deze toestand kan formeel worden gerepresenteerd door een functie v1 over een verzameling van twee elementen, zeg over {MEDEW, AFD}, zodanig dat

v1 (MEDEW) = T1 en v1(AFD) = T2.

Als verder g1 de functie over {AFD, MEDEW} is die wordt gedefinieerd door

g1(AFD) = {ANR, MANNR, NAAM} en
g1(MEDEW) = {AFDNR, GESL, NAAM, NR, SAL},

dan noemen we v1 een *database-toestand* (of kortweg DB-toestand) *over* g1.

□ Voorbeeld 1.2.

We zullen het begrip DB-toestand definiëren voor een willekeurige verzamelingsfunctie g:

DEFINITIE 1.2:
Als g een verzamelingsfunctie is, dan:
v is een **DB-toestand over** $g \overset{D}{\Longleftrightarrow}$ v is een functie over dom(g) en
$\forall E \in \text{dom}(g) : v(E)$ is een tabel over $g(E)$.

Omdat elke DB-toestand een functie is, is elk begrip dat is gedefinieerd voor functies ook van toepassing op DB-toestanden. Zo kunnen we bijvoorbeeld spreken over het domein en het bereik van een DB-toestand.

OPGAVEN

1.3. Wat is het domein en wat is het bereik van de DB-toestand v1 uit Voorbeeld 1.2?

1.4. Kan een DB-toestand over meer dan één verzamelingsfunctie gedefinieerd zijn? (Denk aan de lege tabel.)

□

1.3 DATABASE-UNIVERSA

De DB-toestand v1 genoemd in Voorbeeld 1.2 representeert de toestand van de betreffende organisatie *op een bepaald tijdstip*. Als we aannemen dat we steeds in dezelfde kenmerken geïnteresseerd blijven, dan kan de toestand van de organisatie in kwestie *altijd* worden gerepresenteerd door een DB-toestand over g1. Omgekeerd representeert niet elke DB-toestand over g1 een toegestane toestand voor die organisatie.

De (door de betreffende organisatie zelf te bepalen) verzameling toegestane toestanden is dus één of andere verzameling DB-toestanden over g1. Zo'n verzameling noemen we een *database-universum* (of kortweg DB-universum) *over* g1. In het algemeen:

DEFINITIE 1.3:
Als g een verzamelingsfunctie is, dan:
U is een **DB-universum over** g $\overset{D}{\Longleftrightarrow}$ U is een verzameling DB-toestanden over g.

Als U een DB-universum over g is, dan noemen we g *het DB-skelet van* U, een element E van dom(g) een *tabelindex* (of "tabelnaam") *van* U, $g(E)$ *de kop* (of "heading") *van* E *in* U en een element van $g(E)$ noemen we wel een *attribuut* (of "attribuutnaam") *van* E *in* U. Een element van U noemen we een *DB-toestand conform* U.

VOORBEELD 1.3:

We zullen voor de in Voorbeeld 1.2 genoemde organisatie een DB-universum VBU over g1 definiëren waarin de volgende eisen zijn verwerkt:

(1) Elke medewerker wordt eenduidig bepaald door het medewerkersnummer.

(2) Elke afdeling wordt eenduidig bepaald door het afdelingsnummer.

(3) Elke afdeling wordt ook eenduidig bepaald door de afdelingsnaam.

(4) Elk afdelingsnummer in de medewerkerstabel komt ook voor (als afdelingsnummer) in de afdelingentabel.

(5) Elk managersnummer in de afdelingentabel komt ook voor als medewerkersnummer in de medewerkerstabel.

Om ons DB-universum VBU te definiëren, maken we gebruik van de verzamelingswaardige functies FM en FA uit het vorige hoofdstuk, van twee tabellenverzamelingen WM en WA (die achtereenvolgens de toegestane medewerkerstabellen en de toegestane afdelingentabellen weergeven) en van een hulpfunctie HF (die vastlegt welke tabelindexen aan welke tabellenverzamelingen worden gekoppeld):

$$
\begin{aligned}
\mathrm{FM} = \{ & (\mathrm{NR} && ; \mathbb{N}), \\
& (\mathrm{NAAM} && ; \mathrm{Chs}(40)), \\
& (\mathrm{SAL} && ; \mathbb{N}), \\
& (\mathrm{GESL} && ; \{\text{‘M’}, \text{‘V’}\}), \\
& (\mathrm{AFDNR} && ; [1 \,..\, 99])\};
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{FA} = \{ & (\mathrm{ANR} && ; \mathbb{N}), \\
& (\mathrm{NAAM} && ; \mathrm{Chs}(45)), \\
& (\mathrm{MANNR} && ; \mathbb{N})\};
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{WM} = \{ & T \mid T \subseteq \Pi(\mathrm{FM}) \text{ en} \\
& \forall t \in T : \forall t' \in T : \text{ als } t \neq t' \text{ dan } t(\mathrm{NR}) \neq t'(\mathrm{NR})\}; && (1)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{WA} = \{ & T \mid T \subseteq \Pi(\mathrm{FA}) \text{ en} \\
& \forall t \in T : \forall t' \in T : \text{ als } t \neq t' \text{ dan } t(\mathrm{ANR}) \neq t'(\mathrm{ANR}) \text{ en} && (2) \\
& \qquad\qquad\qquad\qquad\qquad\qquad t(\mathrm{NAAM}) \neq t'(\mathrm{NAAM})\}; && (3)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{HF} = \{ & (\mathrm{MEDEW} && ; \mathrm{WM}), \\
& (\mathrm{AFD} && ; \mathrm{WA})\};
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{VBU} = \{ & v \mid v \in \Pi(\mathrm{HF}) \text{ en} \\
& \{m(\mathrm{AFDNR}) \mid m \in v(\mathrm{MEDEW})\} \subseteq \{a(\mathrm{ANR}) \mid a \in v(\mathrm{AFD})\} \text{ en} && (4) \\
& \{a(\mathrm{MANNR}) \mid a \in v(\mathrm{AFD})\} \subseteq \{m(\mathrm{NR}) \mid m \in v(\mathrm{MEDEW})\}\} && (5)
\end{aligned}
$$

□ Voorbeeld 1.3.

Omdat elk DB-universum een (speciale) verzameling is, is elk begrip dat is gedefinieerd voor verzamelingen ook van toepassing op DB-universa. Bovendien is een DB-universum over een verzamelingsfunctie g formeel gezien tevens een tabel over dom(g):

STELLING 1.1:
Als g een verzamelingsfunctie is en U is een DB-universum over g,
dan is U een tabel over dom(g).

Bewijs:
Uit Definitie 1.3 volgt dat U een verzameling is en
uit Definitie 1.2 volgt dat elk element van U een functie over dom(g) is.
Dus U voldoet aan Definitie 1.1.

□

De tabelindexen van U fungeren dus als de attributen van de "supertabel" U, de DB-toestanden als de tupels van die tabel en de tabellen in zo'n DB-toestand als de "attribuutwaarden" in de tabel U. Zo is bijvoorbeeld VBU een tabel over {MEDEW, AFD}.

OPGAVEN

1.5. Voor een DB-universum U over een verzamelingsfunctie g met $E \in \mathrm{He}(U)$ definiëren we:

(a) **U is consistent** $\overset{D}{\Longleftrightarrow} U \neq \varnothing$;

(b) **E is consistent in U** $\overset{D}{\Longleftrightarrow} \exists v \in U : v(E) \neq \varnothing$;

(c) **U is regulier** $\overset{D}{\Longleftrightarrow}$ U is consistent en
$\forall E' \in \mathrm{He}(U) : E'$ is consistent in U.

Bewijs nu de volgende beweringen:

(a) Als U consistent is, dan is $\mathrm{dom}(g)$ eenduidig bepaald.

(b) Als E consistent in U is, dan is $g(E)$ eenduidig bepaald.

(c) Als U regulier is, dan is g eenduidig bepaald.

Met "eenduidig bepaald" bedoelen we hier achtereenvolgens dat

(a) $\mathrm{dom}(g) = \mathrm{dom}(g')$, (b) $g(E) = g'(E)$, en (c) $g = g'$

voor elke verzamelingsfunctie g' waarover U ook een DB-universum is.

1.6. Bewijs dat er in Opgave 1.5 telkens "dan en slechts dan" geldt.

1.7. Geldt er voor elk DB-universum U dat
U is regulier $\Longleftrightarrow \exists v \in U : \forall E \in \mathrm{He}(U) : v(E) \neq \varnothing$?
(Geef voor elk van beide implicaties een bewijs dan wel een tegenvoorbeeld.)

1.8. Ga na dat $\{\varnothing\}$ een DB-universum is. Is $\{\varnothing\}$ (formeel gezien) een regulier DB-universum? Zo ja, wat is dan het (eenduidig bepaalde) DB-skelet van $\{\varnothing\}$?

1.9. Als U een niet-leeg DB-universum over g is, dan $\mathrm{dom}(g) = \mathrm{He}(U)$. Bewijs dit.

1.10. (a) Is de functie v1 uit Voorbeeld 1.2 een element van VBU?

(b) Geef (nog) een element van VBU.

(c) Is $\varnothing$ een element van WM? En van WA?

(d) Zijn er elementen v van VBU waarvoor geldt $v(\mathrm{AFD}) = \varnothing$?
Zo ja, welke zijn het?

1.11. Een verkooporganisatie wil gegevens bijhouden van haar *klanten*, van de *artikelen* die de organisatie te koop heeft, en van de *orders* die deze klanten voor deze artikelen bij deze organisatie plaatsen.

Van elke klant wil men het klantnummer en de naam van die klant bijhouden. Elke klant wordt eenduidig bepaald door zijn klantnummer. Een klantnummer is een 6-cijferig getal; om precies te zijn: een element van Vng(6).

Van elk artikel wil men het artikelnummer, de naam en de prijs bijhouden. Elk artikel wordt eenduidig bepaald door het (5-cijferige) artikelnummer.

Van elke order wil men de volgende gegevens bijhouden: het ordernummer, het nummer van de klant die de order heeft geplaatst, het nummer van het in de order genoemde artikel en het bestelde aantal exemplaren van dat artikel. (Dus per order wordt er slechts één artikel besteld, maar eventueel met meer exemplaren tegelijk.) Elke order wordt eenduidig bepaald door het (7-cijferige) ordernummer. Van elke order moet het klantnummer ook voorkomen in de klantentabel. Evenzo moet van elke order het artikelnummer in het artikelenbestand voorkomen.

Verwerk deze informele (en niet geheel volledige) beschrijving tot de formele definitie van een DB-universum VBU2.

1.12. Probeer zelf een (klein) DB-universum VBU3 voor een of andere organisatie te definiëren (bijvoorbeeld voor een sportvereniging of een onderwijsinstelling).

1.13. We kunnen elke relatie R als volgt "converteren" naar een tabel over $\{1,2\}$:

$$\mathrm{crt}(R) \overset{D}{=} \{\{(1\,;x)\,,(2\,;y)\} \mid (x\,;y) \in R\}$$

Omgekeerd kunnen we elke tabel T over $\{1,2\}$ "converteren" naar een relatie:

$$\mathrm{ctr}(T) \overset{D}{=} \{(t(1)\,;t(2)) \mid t \in T\}$$

Uit de geconverteerde van een tabel over $\{1,2\}$, respectievelijk een relatie, kunnen we de oorspronkelijke tabel, respectievelijk relatie, op eenduidige wijze reconstrueren door "de andere" conversie toe te passen:

(a) $\mathrm{crt}(\mathrm{ctr}(T)) = T$ voor elke tabel T over $\{1,2\}$;
(b) $\mathrm{ctr}(\mathrm{crt}(R)) = R$ voor elke relatie R.

Bewijs (a) en (b).

□

2 ENIGE CENTRALE DATABASE-BEGRIPPEN

2.1 BEGRIPPEN BETREFFENDE TABELLEN

2.1.1 Operaties op tabellen

We zijn geïnteresseerd in die operaties die aan een of meer tabellen weer een tabel toevoegen.

Een tabel is een verzameling, dus alle operaties die toepasbaar zijn op verzamelingen zijn ook toepasbaar op tabellen. Vele van deze operaties voegen aan elke tabel ook weer een tabel toe (en kunnen dus herhaald worden toegepast). Enkele van deze operaties zullen we hier behandelen.

In het algemeen is elke deelverzameling van een tabel over A weer een tabel over A (Lemma 2.1); in het bijzonder zijn de doorsnede en het verschil van twee tabellen over A weer tabellen over A (Lemma 2.2). Ook is de vereniging van twee tabellen over A weer een tabel over A (Lemma 2.3).

LEMMA 2.1:
Als A een verzameling is en T is een tabel over A en $D \subseteq T$, dan is ook D een tabel over A.

Bewijs:
T is een tabel over A en $D \subseteq T$, dus D is een verzameling en als $t \in D$ dan $t \in T$, dus volgens Definitie 1.1 is dan t een functie over A. We zien dus dat naast T ook D aan Definitie 1.1 voldoet.

□

LEMMA 2.2:
Als A een verzameling is en T en T' zijn tabellen over A, dan zijn ook $T \cap T'$ en $T - T'$ tabellen over A.

Bewijs:
$T \cap T' \subseteq T$ en $T - T' \subseteq T$, dus volgens Lemma 2.1 zijn $T \cap T'$ en $T - T'$ ook tabellen over A.

□

LEMMA 2.3:
Als A een verzameling is en T en T' zijn tabellen over A, dan is ook $T \cup T'$ een tabel over A.

Bewijs:
$T \cup T'$ is een verzameling en als $t \in T \cup T'$ dan $t \in T$ of $t \in T'$; in beide gevallen is t echter een functie over A. Kortom, $T \cup T'$ voldoet aan Definitie 1.1.

□

VOORBEELD 2.0:

Voorbeelden van voornoemde constructies zijn (uitgaande van T1 en T2 uit Voorbeeld 1.1):

(a) $\{t \in \text{T1} \mid t(\text{NR}) \notin \{a(\text{MANNR}) \mid a \in \text{T2}\}\}$;

(b) $\{t \in \text{T1} \mid t(\text{SAL}) > 2500 \text{ en } t(\text{AFDNR}) = 1\}$, ofwel
$\{t \in \text{T1} \mid t(\text{SAL}) > 2500\} \cap \{t \in \text{T1} \mid t(\text{AFDNR}) = 1\}$;

(c) $\{t \in \text{T1} \mid t(\text{NAAM}) \neq \text{'JANSSEN'}\}$, ofwel
$\text{T1} - \{t \in \text{T1} \mid t(\text{NAAM}) = \text{'JANSSEN'}\}$;

(d) $\{t \in \text{T1} \mid t(\text{GESL}) = \text{'V'} \text{ of } t(\text{AFDNR}) = 1\}$, ofwel
$\{t \in \text{T1} \mid t(\text{GESL}) = \text{'V'}\} \cup \{t \in \text{T1} \mid t(\text{AFDNR}) = 1\}$.

□ Voorbeeld 2.0.

Uit Lemma 2.1 blijkt dat het wegstrepen van een aantal "rijen" uit een tabel weer een tabel oplevert. Uit Lemma 2.4 zal blijken dat ook de "orthogonale" operatie van het wegstrepen van een aantal "kolommen" uit een tabel weer een tabel oplevert.

De formele definitie van deze operatie luidt als volgt:

DEFINITIE 2.1:
Als T een verzameling functies is en B is een verzameling, dan:

$T \upharpoonright B \stackrel{D}{=} \{t \upharpoonright B \mid t \in T\}$.

We noemen $T \upharpoonright B$ wel *de projectie van T op B*. Hoewel we $T \upharpoonright B$ hebben gedefinieerd voor *elke* verzameling functies en voor *elke* verzameling B, zijn de interessante gevallen die waarin T een tabel over een verzameling A is en bovendien $B \subseteq A$.

VOORBEELD 2.1:

Figuur 2.1 representeert de projectie van de tabel T1 van Figuur 1.1(a) op de verzameling {GESL, AFDNR}. Merk op dat T1 $\upharpoonright$ {AFDNR, GESL} minder elementen heeft dan T1.

AFDNR	GESL
1	M
2	V

Figuur 2.1: De tabel T1 $\upharpoonright$ {AFDNR, GESL}

□ Voorbeeld 2.1.

LEMMA 2.4:

Als A en B verzamelingen zijn en T is een tabel over A, dan:

(a) $T \upharpoonright B$ is een tabel over $A \cap B$;

(b) $|T \upharpoonright B| \leq |T|$.

Bewijs:

(a) $T \upharpoonright B = \{t \upharpoonright B \mid t \in T\}$ en dus (volgens Lemma 0.11(a)) een verzameling functies over $B \cap A$; met andere woorden, $T \upharpoonright B$ is een tabel over $A \cap B$.

(b) Volgens Lemma 0.8(a) is het voldoende om een functie f met T als domein en $T \upharpoonright B$ als bereik aan te geven. We nemen $f = \lambda t \in T : t \upharpoonright B$; het is duidelijk dat deze functie voldoet.

□

De volgende operatie wordt wel de *natural join* (letterlijk: "natuurlijke verbinding") genoemd:

DEFINITIE 2.2:

Als T en T' verzamelingen functies zijn dan:

$T \bowtie T' \overset{D}{=} \{t \cup t' \mid t \in T \text{ en } t' \in T' \text{ en } t \cup t' \text{ is een functie}\}$.

We kwantificeren hier dus over alle paren $(t; t') \in T \times T'$ waarvoor geldt dat t en t' compatibel zijn (zie Definitie 0.7).

VOORBEELD 2.2:

Figuur 2.2 representeert

(a) een tabel V1 over $\{a, b, c\}$ met 4 elementen,

(b) een tabel V2 over $\{b, c, d\}$ met 5 elementen, en

(c) V1 $\bowtie$ V2, een tabel over $\{a, b, c, d\}$ bestaande uit 7 elementen.

a	b	c
1	1	2
2	1	2
3	2	1
4	2	2

(a) V1

b	c	d
1	1	2
1	2	2
1	2	3
1	2	4
2	2	5

(b) V2

a	*b*	*c*	*d*
1	1	2	2
1	1	2	3
1	1	2	4
2	1	2	2
2	1	2	3
2	1	2	4
4	2	2	5

(c) V1 $\bowtie$ V2

Figuur 2.2

Dus, populair gezegd, elk element van V1 wordt "gecombineerd" met *elk* element van V2 dat er op $\{b, c\}$ "net zo uitziet". Merk op dat elementen die in de andere verzameling geen "gelijkwaardige" tegenhanger hebben in het eindresultaat dus niet voorkomen.

□ Voorbeeld 2.2.

Het volgende lemma geeft enige interessante eigenschappen van de natural join weer:

LEMMA 2.5:
Als A, A' en A'' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en T'' is een tabel over A'', dan:

(a) $T \bowtie T'$ een tabel over $A \cup A'$;

(b) $|T \bowtie T'| \leq |T| * |T'|$;

(c) $T \bowtie T' = T' \bowtie T$, met andere woorden, de natural join is *commutatief*;

(d) $T \bowtie (T' \bowtie T'') = (T \bowtie T') \bowtie T''$, m.a.w. de natural join is *associatief*;

(e) als $A = A'$, dan $T \bowtie T' = T \cap T'$;

(f) $T \bowtie T = T$ en $T \bowtie \{\varnothing\} = T$ en $T \bowtie \varnothing = \varnothing$.

Bewijs:

(a) Uit de definitie van de natural join is het duidelijk dat $T \bowtie T'$ een verzameling functies is. Als $u \in T \bowtie T'$, dan zijn er $t \in T$ en $t' \in T'$ met $u = t \cup t'$;
dus $\mathrm{dom}(u) = \mathrm{dom}(t \cup t') = \mathrm{dom}(t) \cup \mathrm{dom}(t') = A \cup A'$.

(b) We definiëren $f = \{((t\,;t')\,;\ t \cup t') \mid (t\,;t') \in T \times T'$ en $t \cup t'$ is een functie$\}$;
f is een functie en $T \bowtie T' = \mathrm{rng}(f)$ en $\mathrm{dom}(f) \subseteq T \times T'$;
dus, mede dankzij Lemma 0.8(a):
$|T \bowtie T'| = |\mathrm{rng}(f)| \leq |\mathrm{dom}(f)| \leq |T \times T'| = |T| * |T'|$.

(c) $T \bowtie T'$
$= \{t \cup t' \mid t \in T$ en $t' \in T'$ en $t \cup t'$ is een functie$\}$
$= \{t' \cup t \mid t' \in T'$ en $t \in T$ en $t' \cup t$ is een functie$\}$
$= T' \bowtie T$.

(d) $T \bowtie (T' \bowtie T'')$
$= \{t \cup x \mid t \in T$ en $x \in T' \bowtie T''$ en $t \cup x$ is een functie$\}$
$= \{t \cup (t' \cup t'') \mid t \in T$ en $t' \in T'$ en $t'' \in T''$ en
$t' \cup t''$ is een functie en $t \cup (t' \cup t'')$ is een functie$\}$; (1)

$(T \bowtie T') \bowtie T''$
$= \{y \cup t'' \mid y \in T \bowtie T'$ en $t'' \in T''$ en $y \cup t''$ is een functie$\}$
$= \{(t \cup t') \cup t'' \mid t \in T$ en $t' \in T'$ en $t'' \in T''$ en
$t \cup t'$ is een functie en $(t \cup t') \cup t''$ is een functie$\}$; (2)

uit Lemma 0.3(a) blijkt dat in (1) en (2) de voorlaatste eis volgt uit de laatste eis en dus kan worden weggelaten.
Daarna is het eenvoudig in te zien dat $T \bowtie (T' \bowtie T'') = (T \bowtie T') \bowtie T''$.

De rest van het bewijs wordt (in Opgave 2.8) aan de lezer overgelaten.

□

Om het praktische gebruik van joins te illustreren geven we nog een extra voorbeeld van een join.

VOORBEELD 2.3:

Als in Figuur 1.1 (a) van Voorbeeld 1.1 AFDNR wordt vervangen door ANR en NAAM door MEDEWNM dan ontstaat (de representatie van) een tabel T3. Volgens Lemma 2.5 (a) is nu T3 $\bowtie$ T2 een tabel over {NR, MEDEWNM, SAL, GESL, ANR, NAAM, MANNR}. Deze tabel heeft drie elementen en wordt weergegeven in Figuur 2.3.

NR	MEDEWNM	SAL	GESL	ANR	NAAM	MANNR
8	JANSSEN	2200	M	1	PRODUCTIE	9
7	JANSSEN	3109	V	2	PLANNING	7
9	DEKKER	2995	M	1	PRODUCTIE	9

Figuur 2.3: De tabel T3 $\bowtie$ T2

Kennelijk bevat T3 $\bowtie$ T2 voor elke medewerker de gegevens van die medewerker *plus* de gegevens van diens afdeling.

□ Voorbeeld 2.3.

In Voorbeeld 2.3 beschreven we op informele wijze een manier om uit een tabel een andere tabel te verkrijgen, namelijk door de "kolomnamen" te vervangen door andere kolomnamen. De vraag is nu hoe we deze (nuttige) operatie op tabellen formeel kunnen definiëren. Daartoe maken we gebruik van een "herbenoemingsfunctie" (of *attributentransformatie*) *h* die aan elke "nieuwe" kolomnaam *b* de door *b* vervangen oude kolomnaam toevoegt:

DEFINITIE 2.3:

Als T een verzameling van functies is en h is een functie, dan:

$T \infty h \overset{D}{=} \{t \circ h \mid t \in T\}$.

VOORBEELD 2.4:

De in Voorbeeld 2.3 gebruikte attributentransformatie is de als volgt gedefinieerde functie h3 over {NR, MEDEWNM, SAL, GESL, ANR}:

h3(ANR) = AFDNR,
h3(MEDEWNM) = NAAM en
h3(x) = x voor elke $x \in$ {NR, SAL, GESL}.

We merken op dat h3 een *bijectie* op {NR, NAAM, SAL, GESL, AFDNR} is, dat is precies de verzameling waarover T1 een tabel is. De in Voorbeeld 2.3 beschreven tabel T3 kunnen we nu schrijven als T1 ∞ h3. Ter illustratie bekijken we het effect voor één van de elementen van T1 nog eens in detail; als t3 het element van T1 is waarvoor geldt t3(NR) = 9, dan zien we dat t3 $\circ$ h3 inderdaad het gewenste gedrag heeft:

t3 $\circ$ h3(ANR) = t3(h3(ANR)) = t3(AFDNR) = 1.

Figuur 2.4 geeft het voorgaande nog eens suggestief weer.

NR	MEDEWNM	SAL	GESL	ANR
NR	NAAM	SAL	GESL	AFDNR

(a) De attributentransformatie h3

NR	NAAM	SAL	GESL	AFDNR
9	DEKKER	2995	M	1
8	JANSSEN	2200	M	1
7	JANSSEN	3109	V	2

(b) De tabel T1

NR	MEDEWNM	SAL	GESL	ANR
9	DEKKER	2995	M	1
8	JANSSEN	2200	M	1
7	JANSSEN	3109	V	2

(c) De tabel T1 ∞ h3

Figuur 2.4

□ Voorbeeld 2.4.

We benadrukken nog eens dat de attributentransformatie de oude attributen aan de nieuwe attributen toevoegt, en niet omgekeerd.

Definitie 2.3 laat de mogelijkheid open om één "oud" attribuut te vervangen door twee of meer "nieuwe" attributen. Uit de definitie volgt dat het resultaat in dat geval is dat er als het ware "kopieën" van de oude kolom onder elk van die nieuwe attributen voorkomen. Deze

nieuwe attributen zijn dus eigenlijk "synoniemen" van elkaar.

Lemma 2.6 bevestigt ons vermoeden dat herbenoeming van attributen in een tabel weer een tabel met evenveel elementen oplevert.

LEMMA 2.6:
Als h een functie is en T is eèn tabel over rng(h), dan:

(a) $T \infty h$ is een tabel over dom(h);

(b) $| T \infty h | = | T |$;

(c) $T \infty h = \varnothing$ desda $T = \varnothing$.

Bewijs:

(a) Uit Lemma 0.9, (a) en (d), blijkt dat voor elke $t \in T$ geldt dat $t \circ h$ een functie is en, omdat rng(h) = dom(t), dat dom($t \circ h$) = dom(h). Dus $T \infty h$ is een tabel over dom(h).

(b) Het is eenvoudig in te zien dat $\lambda t \in T$: $t \circ h$ een functie van T op $T \infty h$ is.
De functie is tevens injectief:
Stel dat $t \in T$ en $t' \in T$ zodanig dat $t \circ h = t' \circ h$; voor elke $y \in \text{dom}(t)$ geldt dat $y \in \text{rng}(h)$, met andere woorden, er is een $x \in \text{dom}(h)$ met $h(x) = y$;
dus $t(y) = t(h(x)) = t'(h(x)) = t'(y)$. Uit Lemma 0.4(b) volgt nu dat $t = t'$.
Kortom, de functie $\lambda t \in T$: $t \circ h$ van T op $T \infty h$ is ook injectief (zie Lemma 0.6).

(c) Dit volgt direct uit (b) omdat $\varnothing$ de enige verzameling is met 0 elementen.

□

OPGAVEN

2.1. (a) Kunnen we, de Lemma's 2.2 en 2.3 samenvattend, zeggen dat zowel de vereniging als de doorsnede en het verschil van elk tweetal tabellen weer een tabel is?

(b) Geef de vier in Voorbeeld 2.0 beschreven verzamelingen medewerkerstupels in woorden weer.

2.2. Als A een verzameling is en W is een verzameling van tabellen over A, dan is ook $\bigcup W$ een tabel over A. Bewijs dit.

2.3. Leg uit waarom T1 $\upharpoonright$ {AFDNR,GESL} minder elementen heeft dan T1.
(Vergelijk bijvoorbeeld Figuur 2.1 met Figuur 2.4 (b).)

2.4. Als T een verzameling functies is en B is een verzameling, dan:

(a) $T \upharpoonright B = \varnothing$ desda $T = \varnothing$;

(b) $T \upharpoonright B = \{\varnothing\}$ desda $T \neq \varnothing$ en $B \cap \mathrm{He}(T) = \varnothing$.

Bewijs dit.

2.5. Geldt (b) van Lemma 2.4 ook als T een willekeurige verzameling functies is? (Geef een bewijs of een tegenvoorbeeld.)

2.6. Bewijs dat voor alle verzamelingen A, B en B' en voor alle tabellen T en T' over A het volgende geldt:

(a) $(T \cup T') \upharpoonright B = (T \upharpoonright B) \cup (T' \upharpoonright B)$;

(b) $(T \cap T') \upharpoonright B \subseteq (T \upharpoonright B) \cap (T' \upharpoonright B)$;

(c) $(T - T') \upharpoonright B \supseteq (T \upharpoonright B) - (T' \upharpoonright B)$;

(d) $(T \upharpoonright B') \upharpoonright B = T \upharpoonright (B' \cap B)$.

Geef voor de twee "ontbrekende" inclusies tegenvoorbeelden.

2.7. De formule in Opgave 2.6(a) kan aldus worden verwoord: "Projectie is *rechtsdistributief* over vereniging".

(a) Is projectie ook "linksdistributief" over vereniging? (Geef eerst aan wat we hiermee zouden bedoelen.)

(b) Is projectie linksdistributief over doorsnede of over verschil?

2.8. Bewijs zelf de rest van Lemma 2.5.

2.9. Als in Figuur 1.1(b) MANNR wordt vervangen door NR en NAAM door ANM dan ontstaat (de representatie van) een tabel T4.

(a) Bepaal T1 $\bowtie$ T4.

(b) Bepaal T3 $\bowtie$ T4.

(c) Bepaal T1 $\bowtie$ T5 en T3 $\bowtie$ T5, waarbij T5 = T4 $\cup$ {t5}, waarbij t5 = {(ANR; 3), (ANM; 'INKOOP'), (NR; 7)}.

(d) Geef een informele beschrijving van wat de tabellen T1 $\bowtie$ T5 en T3 $\bowtie$ T5 achtereenvolgens voorstellen.

2.10. Bepaal de natural join van T1 en T2 uit Voorbeeld 1.1. Probeer ook een informele beschrijving te geven van wat T1 $\bowtie$ T2 kennelijk "voorstelt".

2.11. Bewijs dat voor alle verzamelingen A en A' en voor alle tabellen T over A en T' over A' het volgende geldt:

(a) $(T \bowtie T') \upharpoonright A \subseteq T$;

(b) als $T \upharpoonright A' \subseteq T' \upharpoonright A$, dan $(T \bowtie T') \upharpoonright A = T$;

(c) als $A' \subseteq A$, dan $T \bowtie T' = \{t \in T \mid t \upharpoonright A' \in T'\}$;

(d) als $A \cap A' = \varnothing$, dan $\mid T \bowtie T' \mid = \mid T \mid * \mid T' \mid$.

2.12. Gebruik de resultaten uit de vorige opgave om te bewijzen dat voor elk element v van het DB-universum VBU uit Voorbeeld 1.3 het onderstaande geldt. Hierbij is h3 de attributentransformatie die in Voorbeeld 2.4 aldus is gedefinieerd:

h3(ANR) = AFDNR, h3 (MEDEWNM) = NAAM en
h3(x) = x voor elke $x \in$ {NR, SAL, GESL}.

Verder geldt voor het in Voorbeeld 1.2 geïntroduceerde DB-skelet g1 dat g1(AFD) = {ANR, MANNR, NAAM}.

(a) $(v\,(\text{AFD}) \bowtie (v\,(\text{MEDEW}) \infty \text{h3})) \upharpoonright \text{g1(AFD)} \quad \subseteq v(\text{AFD})$;

(b) $(v\,(\text{AFD}) \bowtie (v(\text{MEDEW}) \infty \text{h3})) \upharpoonright \text{dom(h3)} \quad = v(\text{MEDEW}) \infty \text{h3}$;

(c) $v\,(\text{AFD}) \bowtie (v(\text{MEDEW}) \infty \{(\text{MANNR; NR})\}) \quad = v(\text{AFD})$.

2.13. Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$,
dan $T \upharpoonright (B \cup C) \subseteq (T \upharpoonright B) \bowtie (T \upharpoonright C)$.
Bewijs dit.

2.14. Bewijs dat voor alle verzamelingen A en A' en voor alle tabellen T over A en T' en T'' over A' het volgende geldt:

(a) $T \bowtie (T' \cap T'') = (T \bowtie T') \cap (T \bowtie T'')$;

(b) $T \bowtie (T' - T'') = (T \bowtie T') - (T \bowtie T'')$;

(c) $T \bowtie (T' \cup T'') = (T \bowtie T') \cup (T \bowtie T'')$.

2.15. Opgave 2.14 kan aldus worden verwoord: "De natural join is *linksdistributief* over doorsnede, verschil en vereniging".

(a) Is de natural join ook "rechtsdistributief" over verschil?
(Formuleer eerst wat we hiermee zouden bedoelen.)

(b) Is de natural join ook rechtsdistributief over doorsnede en over vereniging?

2.16. Geef een formele definitie van de in Opgave 2.9 gebruikte attributentransformatie, dat wil zeggen een functie h2 zodanig dat T4 = T2 ∞ h2 (waarbij T2 de in Figuur 1.1 (b) gerepresenteerde tabel is).

2.17. Als T een verzameling functies is en h en h' zijn functies, dan:

(a) $T \infty (h \circ h') = (T \infty h) \infty h'$;

(b) $T \infty (h \cup h') \subseteq (T \infty h) \bowtie (T \infty h')$ als h en h' compatibel zijn.

Bewijs dit.

2.18. Bewijs dat voor alle functies h en voor alle tabellen T en T' over rng(h) het volgende geldt:

(a) $(T \cup T') \infty h = (T \infty h) \cup (T' \infty h)$;

(b) $(T \cap T') \infty h = (T \infty h) \cap (T' \infty h)$;

(c) $(T - T') \infty h = (T \infty h) - (T' \infty h)$;

(d) als $T \subseteq T'$ dan $T \infty h \subseteq T' \infty h$;

(e) als h injectief is, dan:
$T \subseteq T' \Leftrightarrow T \infty h \subseteq T' \infty h$.

2.19. Bewijs het volgende:
Als T een verzameling functies is en B is een verzameling en h is een functie, dan:

(a) $T \upharpoonright B = T \infty \mathrm{id}(B)$;
(b) $(T \infty h) \upharpoonright B = T \infty (h \upharpoonright B)$;
(c) $(T \upharpoonright B) \infty h = T \infty (\mathrm{id}(B) \circ h)$.

2.20. Bewijs het volgende:
Als A en A' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en h is een functie, dan:

(a) $(T \bowtie T') \infty h \subseteq (T \infty h) \bowtie (T' \infty h)$;

(b) $(T \bowtie T') \infty h = (T \infty h) \bowtie (T' \infty h)$ als $A \cap A' \subseteq \mathrm{rng}(h)$.

2.21. Leid uit de vorige opgave het volgende af:
Als A en A' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en B is een verzameling, dan:

(a) $(T \bowtie T') \upharpoonright B \subseteq (T \upharpoonright B) \bowtie (T' \upharpoonright B)$;

(b) $(T \bowtie T') \upharpoonright B = (T \upharpoonright B) \bowtie (T' \upharpoonright B)$ als $A \cap A' \subseteq B$.

2.22. Als T een verzameling functies is en B is een verzameling, dan definiëren we:

$$T \mathbin{\not\upharpoonright} B \overset{D}{=} \{t \mathbin{\not\upharpoonright} B \mid t \in T\}.$$

Gebruik nu Lemma 0.11 en de eigenschappen van projectie als leidraad om enige eigenschappen van dit nieuwe begrip af te leiden.

□

2.1.2 Momentane afhankelijkheid

Als T een tabel over A is en B en C zijn deelverzamelingen van A, dan noemen we C *momentaan afhankelijk van B in T* desda elk tweetal elementen van T dat overeenstemt op B ook overeenstemt op C. We noteren dit kortheidshalve als volgt: $B \rightarrow C$ in T. (Deze notatie dient niet te worden verward met de notatie "$B \twoheadrightarrow C$" uit Definitie 0.10; zie in dit verband ook Opgave 2.25(b).) De formele definitie luidt:

DEFINITIE 2.4:
Als A , B en C verzamelingen zijn en T is een tabel over A, dan:

$$B \rightarrow C \text{ in } T \overset{D}{\Longleftrightarrow} \forall t \in T : \forall t' \in T : \text{ als } t \upharpoonright B = t' \upharpoonright B \text{ dan } t \upharpoonright C = t' \upharpoonright C.$$

Hoewel we momentane afhankelijkheid in een tabel over A hebben gedefinieerd voor alle verzamelingen B en C, zijn de interessante gevallen die waarin $B \subseteq A$ en $C \subseteq A$. (Formeel hebben we de andere gevallen ook niet eens nodig; zie Opgave 2.25(a).)

VOORBEELD 2.5:
Als we de in Voorbeeld 1.1 ingevoerde tabel T1 over {NR, NAAM, SAL, GESL, AFDNR} bekijken (zie Figuur 2.5), dan constateren we onder andere dat:

(a) $\{NR\} \to \{NR, NAAM, SAL, GESL, AFDNR\}$ in T1;

(b) $\{SAL\} \to \{NR, NAAM, SAL, GESL, AFDNR\}$ in T1;

(c) $\{GESL\} \to \{AFDNR\}$ in T1;

(d) $\{AFDNR\} \to \{GESL\}$ in T1;

(e) $\{NAAM, GESL\} \to \{SAL\}$ in T1;

(f) $\{NAAM, AFDNR\} \to \{NR, SAL, GESL\}$ in T1;

(g) niet $\{NAAM\} \to \{NR, SAL, GESL, AFDNR\}$ in T1;

(h) niet $\{AFDNR\} \to \{NR, SAL, GESL, NAAM\}$ in T1.

NR	NAAM	SAL	GESL	AFDNR
7	JANSSEN	3109	V	2
8	JANSSEN	2200	M	1
9	DEKKER	2995	M	1

Figuur 2.5: De tabel T1

□ Voorbeeld 2.5.

We vervolgen met een viertal lemma's; Lemma 2.7 bevat drie belangrijke basiseigenschappen van momentane afhankelijkheid, Lemma 2.8 geeft enige directe gevolgen ervan, Lemma 2.9 behandelt de randgevallen en Lemma 2.10 tenslotte geeft een alternatieve karakterisering van momentane afhankelijkheid.

LEMMA 2.7:
Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$ en $D \subseteq A$, dan:

(a) als $C \subseteq B$, dan $B \to C$ in T;

(b) als $B \to C$ in T en $C \to D$ in T, dan $B \to D$ in T;

(c) $B \to C$ in T desda $\forall c \in C : B \to \{c\}$ in T.

Bewijs:

(a) Stel dat $t \in T$ en $t' \in T$ zodanig dat $t \restriction B = t' \restriction B$;
voor $C \subseteq B$ geldt dan uiteraard ook $t \restriction C = t' \restriction C$;
dus $B \to C$ in T.

(b) $B \to C$ in T, dus $\forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $t \restriction C = t' \restriction C$; (1)
$C \to D$ in T, dus $\forall t \in T : \forall t' \in T$: als $t \restriction C = t' \restriction C$ dan $t \restriction D = t' \restriction D$; (2)
uit (1) en (2) volgt $\forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $t \restriction D = t' \restriction D$;
dus $B \to D$ in T.

(c) $t \restriction C = t' \restriction C$ desda $\forall c \in C : t(c) = t'(c)$;
anders gezegd, $t \restriction C = t' \restriction C$ desda $\forall c \in C : t \restriction \{c\} = t' \restriction \{c\}$; daarom
$B \rightarrow C$ in T
$\Leftrightarrow \forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $\forall c \in C : t \restriction \{c\} = t' \restriction \{c\}$
$\Leftrightarrow \forall c \in C : \forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $t \restriction \{c\} = t' \restriction \{c\}$
$\Leftrightarrow \forall c \in C : B \rightarrow \{c\}$ in T.

□

LEMMA 2.8:
Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$ en $D \subseteq A$ en $E \subseteq A$, dan:

(a) $B \rightarrow B$ in T;

(b) als $B \rightarrow C$ in T en $B \subseteq D$, dan $D \rightarrow C$ in T;

(c) als $B \rightarrow C$ in T en $D \subseteq C$, dan $B \rightarrow D$ in T;

(d) als $B \rightarrow C$ in T en $D \rightarrow E$ in T, dan $B \cup D \rightarrow C \cup E$ in T.

Lemma 2.8 is rechtstreeks met behulp van Lemma 2.7 te bewijzen, dat wil zeggen zonder terug te grijpen op de feitelijke *definitie* van momentane afhankelijkheid. (Sterker nog, in zekere zin karakteriseren de eigenschappen in Lemma 2.7, de zogeheten *Armstrong-axioma's*, precies alle afhankelijkheidsstructuren die in tabellen mogelijk zijn; zie [Ar 74] voor de details.)

Bewijs:

(a) Neem C gelijk aan B in Lemma 2.7 (a).

(b) $B \subseteq D$, dus $D \rightarrow B$ in T volgens Lemma 2.7 (a);
omdat ook $B \rightarrow C$ in T, volgt nu uit Lemma 2.7 (b) dat $D \rightarrow C$ in T.

(c) Analoog aan het bewijs van (b).

(d) $B \rightarrow C$ in T en $B \subseteq B \cup D$, dus $B \cup D \rightarrow C$ in T volgens onderdeel (b);
evenzo $B \cup D \rightarrow E$ in T;
uit Lemma 2.7 (c) volgt nu dat
($\forall c \in C : B \cup D \rightarrow \{c\}$ in T) en ($\forall c \in E : B \cup D \rightarrow \{c\}$ in T);
kortom, $\forall c \in C \cup E : B \cup D \rightarrow \{c\}$ in T;
tenslotte gebruiken we Lemma 2.7 (c) nog een keer om te kunnen concluderen dat
$B \cup D \rightarrow C \cup E$ in T.

□

LEMMA 2.9:

Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$, dan:

(a) $B \rightarrow C$ in $\varnothing$;

(b) $B \rightarrow \varnothing$ in T;

(c) $\varnothing \rightarrow C$ in T desda $| T \upharpoonright C | \leq 1$.

Bewijs:

(a) Volgt direct uit Definitie 2.4.

(b) Neem C gelijk aan $\varnothing$ in Lemma 2.7 (a).

(c) $\varnothing \rightarrow C$ in T

$\Leftrightarrow \forall t \in T : \forall t' \in T$: als $t \upharpoonright \varnothing = t' \upharpoonright \varnothing$ dan $t \upharpoonright C = t' \upharpoonright C$ (volgens Definitie 2.4)

$\Leftrightarrow \forall t \in T : \forall t' \in T$: als $\varnothing = \varnothing$ dan $t \upharpoonright C = t' \upharpoonright C$ (volgens Lemma 0.11(l))

$\Leftrightarrow \forall t \in T : \forall t' \in T : t \upharpoonright C = t' \upharpoonright C$

$\Leftrightarrow \forall x \in T \upharpoonright C : \forall x' \in T \upharpoonright C : x = x'$

$\Leftrightarrow | T \upharpoonright C | \leq 1$.

□

LEMMA 2.10:

Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$, dan:
$B \rightarrow C$ in T desda $\{(t \upharpoonright B ; t \upharpoonright C) \mid t \in T\}$ is een functie.

Bewijs:

$\{(t \upharpoonright B ; t \upharpoonright C) \mid t \in T\}$ is een functie

$\Leftrightarrow \forall t \in T : \forall t' \in T$: als $t \upharpoonright B = t' \upharpoonright B$ dan $t \upharpoonright C = t' \upharpoonright C$ (volgens Definitie 0.6)

$\Leftrightarrow B \rightarrow C$ in T (volgens Definitie 2.4)

□

In de vorige lemma's "varieerden" we meestal de attributenverzamelingen "bij gelijkblijvende T". In het volgende lemma "variëren" we de tabel "bij gelijkblijvende attributenverzameling", om te kijken in hoeverre momentane afhankelijkheid behouden blijft onder de operaties uit § 2.1.1. In de opgaven 2.27 en 2.28 gaan we hierop door.

LEMMA 2.11:

Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$ en $B \rightarrow C$ in T, dan:

(a) als $T' \subseteq T$, dan $B \rightarrow C$ in T';

(b) als B' een verzameling is zodanig dat $B \subseteq B'$, dan $B \rightarrow C$ in $T \Vert\!\upharpoonright B'$;

(c) als A' een verzameling is en T' is een tabel over A', dan $B \rightarrow C$ in $T \bowtie T'$;

(d) als h een functie is en $B' \subseteq \mathrm{dom}(h)$ zodanig dat $B \subseteq \mathrm{rng}(h \upharpoonright B')$,
dan $B' \rightarrow \{a \in \mathrm{dom}(h) \mid h(a) \in C\}$ in $T \infty h$.

OPGAVEN

2.23. Bepaal alle momentane afhankelijkheden in de tabel T1 van Figuur 2.5 die van de vorm $B \rightarrow \{c\}$ zijn, waarbij $c \in$ {NR, NAAM, SAL, GESL, AFDNR} $- B$ en B minimaal is in de volgende zin: $\forall B' \subseteq B$: als $B' \rightarrow \{c\}$ in T1 dan $B' = B$.

2.24. Leg uit waarom de momentane afhankelijkheden van bovengenoemde vorm te zamen in wezen voldoende zijn om alle momentane afhankelijkheden in een tabel over een eindige attributenverzameling te kennen.

2.25. Als A, B en C verzamelingen zijn en T is een tabel over A, dan:

(a) $B \rightarrow C$ in $T \Longleftrightarrow (B \cap A) \rightarrow (C \cap A)$ in T;

(b) $B \rightarrow C$ in $B \rightarrowtriangle C$.

Bewijs dit.

2.26. Bewijs Lemma 2.11.

2.27. Laat aan de hand van een voorbeeld zien dat momentane afhankelijkheid in het algemeen niet behouden blijft onder de vereniging van twee tabellen over een zelfde verzameling A.

2.28. Bewijs het volgende:
Als A en A' verzamelingen en T is een tabel over A en T' is een tabel over A' en $B \subseteq A$ en $C \subseteq A \cap A'$ en $D \subseteq A'$, dan:
als $B \rightarrow C$ in T en $C \rightarrow D$ in T', dan $B \rightarrow D$ in $T \bowtie T'$.

□

2.1.3 Unieke identificatie

Als T een tabel over een verzameling A is en $B \subseteq A$ dan noemen we B *uniek identificerend* (of kortweg *u.i.*) *in T* desda twee verschillende elementen van T altijd verschillende restricties tot B hebben. Unieke identificatie is in feite een (belangrijk) speciaal geval van momentane afhankelijkheid (zie Lemma 2.12). We noemen B *minimaal u.i. in T* desda B u.i. in T is en elke echte deelverzameling van B niet u.i. in T is. De formele definities van deze begrippen luiden als volgt:

DEFINITIE 2.5:
Als A en B verzamelingen zijn en T is een tabel over A, dan:
B is u.i. in T $\overset{D}{\Longleftrightarrow} \forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $t = t'$.

DEFINITIE 2.6:
Als A en B verzamelingen zijn en T is een tabel over A, dan:
B is minimaal u.i. in T $\overset{D}{\Longleftrightarrow}$ B is u.i. in T en
$\forall B' \subset B : B'$ is niet u.i. in T.

VOORBEELD 2.6:
Voor de tabel T1 uit Voorbeeld 2.5 geldt het volgende:

(a) {NAAM, NR} is u.i. in T1;

(b) {NAAM, SAL} is u.i. in T1;

(c) {NAAM} is niet u.i. in T1;

(d) {GESL, AFDNR} is niet u.i. in T1;

(e) elk van de verzamelingen {NR}, {SAL}, {NAAM, GESL} en {NAAM, AFDNR} is minimaal u.i. in T1;

(f) geen enkele andere verzameling (dan de onder (e) genoemde) is minimaal u.i. in T1.

□ Voorbeeld 2.6.

De volgende lemma's sommen enkele eenvoudige eigenschappen van unieke identificatie op. De meeste eigenschappen volgen direct uit de lemma's in § 2.1.2.

LEMMA 2.12:
Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en $C \subseteq A$ en $D \subseteq A$, dan:

(a) B is u.i. in T desda $B \rightarrow A$ in T;

(b) B is u.i. in T desda $\forall a \in A - B : B \rightarrow \{a\}$ in T;

(c) A is u.i. in T;

(d) als B u.i. in T is en $B \subseteq C$, dan is C u.i. in T;

(e) als B u.i. in T is en $C \rightarrow B$ in T, dan is C u.i. in T;

(f) als $B \rightarrow C$ in T en $D \rightarrow (A - C)$ in T, dan is $B \cup D$ u.i. in T;

(g) $\varnothing$ is u.i. in T desda $| T | \leq 1$.

LEMMA 2.13:
Als A een verzameling is en T is een tabel over A en $B \subseteq A$ en B is u.i. in T, dan:

(a) als $T' \subseteq T$, dan is B u.i. in T';

(b) als $B \subseteq B'$, dan is B u.i. in $T \restriction B'$;

(c) als h een functie is en $B \subseteq \mathrm{rng}(h)$, dan is $\mathrm{dom}(h)$ u.i. in $T \infty h$.

LEMMA 2.14:
Als A en A' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en $B \subseteq A$ en $B' \subseteq A'$ en B is u.i. in T en B' is u.i. in T', dan:

(a) $B \cup B'$ is u.i. in $T \bowtie T'$;

(b) als $B' \subseteq A$, dan is B u.i. in $T \bowtie T'$.

We zullen hier alleen Lemma 2.14 (a) bewijzen. De andere bewijzen worden in de opgaven gevraagd.

Bewijs van Lemma 2.14 (a):

B is u.i. in T en B' is u.i. in T',	(gegeven)
dus $B \rightarrow A$ in T en $B' \rightarrow A'$ in T',	(volgens Lemma 2.12 (a))
dus $B \rightarrow A$ in $T \bowtie T'$ en $B' \rightarrow A'$ in $T' \bowtie T$,	(volgens Lemma 2.11 (c))
dus $B \cup B' \rightarrow A \cup A'$ in $T \bowtie T'$,	(volgens Lemma 2.8 (d), en Lemma 2.5 (c))
dus $B \cup B'$ is u.i. in $T \bowtie T'$	(volgens Lemma 2.12 (a), en Lemma 2.5 (a)

□

OPGAVEN

2.29. Herschrijf de definities van WM en WA uit Voorbeeld 1.3 in termen van unieke identificatie.

2.30. Ga voor elk van de volgende tabellen na welke verzamelingen minimaal uniek identificerend in die tabel zijn.

(a) T2 uit Voorbeeld 1.1;

(b) T5 uit Opgave 2.9 (c).

2.31. Bewijs Lemma 2.12.

2.32. Bewijs Lemma 2.13.

2.33. Bewijs Lemma 2.14 (b).

2.34. Momentane afhankelijkheid is ook in termen van unieke identificatie te definiëren, want voor alle tabellen T en alle verzamelingen B en C geldt:

$B \to C$ in $T \Leftrightarrow B$ is u.i. in $T \upharpoonright (B \cup C)$.

Bewijs dit.

2.35. Bewijs onderstaande equivalenties (cf. Opgave 1.13).
Als R een relatie is, dan:

(a) R is een functie $\Leftrightarrow$ $\{1\}$ is u.i. in crt(R);

(b) R is injectief $\Leftrightarrow$ $\{2\}$ is u.i. in crt(R).

□

2.1.4 Het verbindingsbegrip

Het volgende nieuwe begrip leiden we in met een voorbeeld.

VOORBEELD 2.7:

In Voorbeeld 1.1 konden we constateren dat elke "(ANR; MANNR)-combinatie" in de tabel T2 ook voorkomt als "(AFDNR; NR)-combinatie" in de tabel T1, zie ook Figuur

2.6. Anders gezegd, de projectie van T2 op {ANR, MANNR} is "bijna" een deelverzameling van de projectie van T1 op {AFDNR, NR}, namelijk op een herbenoeming van de attributen na. De bedoelde herbenoeming kunnen we weergeven door de volgende attributentransformatie:

$$h4 = \{(ANR;\ AFDNR), (MANNR;\ NR)\}$$

De attributentransformatie h4 voegt dus aan elk der relevante attributen van T2 het "corresponderende" attribuut van T1 toe. Volgens onze definitie van functiecompositie geldt nu dat de projectie van T2 op {ANR, MANNR} wél een deelverzameling van de "herbenoemde" tabel $\{t \circ h4 \mid t \in T1\}$ is. (Dit is namelijk wel een tabel over {ANR, MANNR}.) Met behulp van Definitie 2.3 kunnen we nu onze aanvankelijke constatering als volgt formeel weergeven:

$$T2 \upharpoonright dom(h4) \subseteq T1 \infty h4$$

We zeggen in dit geval wel dat de attributentransformatie h4 de tabel T2 *verbindt met* de tabel T1.

T2:

MANNR	ANR	NAAM
9	1	PRODUCTIE
7	2	PLANNING

MANNR	ANR

h4: ↓ ↓

T1:

NR	AFDNR	NAAM	SAL	GESL
8	1	JANSSEN	2200	M
9	1	DEKKER	2995	M
7	2	JANSSEN	3109	V

Figuur 2.6: De attributentransformatie h4 verbindt T2 met T1

□ Voorbeeld 2.7.

De algemene definitie van het verbindingsbegrip luidt als volgt:

DEFINITIE 2.7:
Als T en T' verzamelingen functies zijn en h is een functie, dan:
h verbindt T met T' $\overset{D}{\Longleftrightarrow} T \Vert \mathrm{dom}(h) \subseteq T' \infty h$.

Twee iets minder compacte formuleringen zijn:
(1) h verbindt T met $T' \Longleftrightarrow \{t \restriction \mathrm{dom}(h) \mid t \in T\} \subseteq \{t' \circ h \mid t' \in T'\}$;
(2) h verbindt T met $T' \Longleftrightarrow \forall t \in T : \exists t' \in T' : t \restriction \mathrm{dom}(h) = t' \circ h$.

In de speciale gevallen waarin wij meestal zijn geïnteresseerd is T een tabel over een verzameling A, T' een tabel over een verzameling A' en h een injectieve functie waarbij $\mathrm{dom}(h) \subseteq A$ en $\mathrm{rng}(h) \subseteq A'$. Zij nu $B = \mathrm{dom}(h)$ en $B' = \mathrm{rng}(h)$, dan kunnen we de volgende informele beschrijving van het verbindingsbegrip geven:

h verbindt T met T' dan en slechts dan als alle B-waarden in T ook voorkomen als B'-waarden in T', waarbij h aangeeft welk attribuut in B correspondeert met welk attribuut in B'.

In de praktijk zal $\mathrm{rng}(h)$ vaak uniek identificerend in T' zijn. Bovendien zijn de attributen in B vaak gelijk aan de corresponderende attributen in B'; in dat geval is h dus de identieke functie op B.

In sommige situaties is er bij verbinding niet alleen sprake van inclusie (zoals in Definitie 2.7) maar zelfs van gelijkheid. We zeggen dan: *h verbindt T bilateraal met T'*. Zo verbindt in Voorbeeld 2.7 de functie {(AFDNR; ANR)} de tabel T1 bilateraal met T2.

DEFINITIE 2.8:
Als T en T' verzamelingen functies zijn en h is een functie dan:
h verbindt T bilateraal met T' $\overset{D}{\Longleftrightarrow} T \Vert \mathrm{dom}(h) = T' \infty h$.

Voor een injectieve functie h is bilaterale verbinding eenvoudig uit te drukken in termen van "gewone" verbinding:

LEMMA 2.15:
Als T en T' verzamelingen functies zijn en h is een injectieve functie, dan:
h verbindt T bilateraal met $T' \Longleftrightarrow h$ verbindt T met T' en h^{-1} verbindt T' met T.

Het bewijs van dit lemma wordt in de opgaven aan de lezer overgelaten.

Als T en T' verzamelingen functies zijn en h is een functie, dan noemen we

$\{(t\,;t') \in T \times T' \mid t \restriction \mathrm{dom}(h) = t' \circ h\}$

de door h geïnduceerde associatie op $T \times T'$ (of tussen T en T').

Informeel verwoord is het dus de verzameling van alle tupelparen die "bij elkaar passen" in de zin van het "criterium" h.

Met behulp van deze geïnduceerde associatie kunnen we nu eenvoudige alternatieve omschrijvingen geven van de begrippen verbinding en bilaterale verbinding, zie het volgende lemma. Tevens blijkt unieke identificatie van $\mathrm{rng}(h)$ in T' of van $\mathrm{dom}(h)$ in T interessante eigenschappen van de geïnduceerde associatie tot gevolg te hebben.

LEMMA 2.16:
Als T en T' verzamelingen functies zijn en h is een functie en R is de door h geïnduceerde associatie op $T \times T'$, dus

$$R = \{(t\,;t') \in T \times T' \mid t \restriction \mathrm{dom}(h) = t' \circ h\}\,,$$

dan:

(0) $R \subseteq T \times T'$;

(1) h verbindt T met T' $\iff \mathrm{dom}(R) = T$;

(2) h verbindt T bilateraal met T' $\iff \mathrm{dom}(R) = T$ en $\mathrm{rng}(R) = T'$;

(3) als $\mathrm{rng}(h)$ u.i. in T' is, dan is R een functie;

(4) als $\mathrm{dom}(h)$ u.i. in T is, dan is R injectief.

Bewijs:

(0) Het volgt direct uit de beschrijving van R dat $R \subseteq T \times T'$.
(In het bijzonder geldt dus dat $\mathrm{dom}(R) \subseteq T$ en $\mathrm{rng}(R) \subseteq T'$.)

(1) h verbindt T met T'
$\iff \{t \restriction \mathrm{dom}(h) \mid t \in T\} \subseteq \{t' \circ h \mid t' \in T'\}$
$\iff \forall t \in T : \exists t' \in T' : t \restriction \mathrm{dom}(h) = t' \circ h$
$\iff T \subseteq \mathrm{dom}(R)$;
(1) volgt nu uit (0) en de voorgaande equivalenties.

(2) h verbindt T bilateraal met T'
$\iff \{t \restriction \mathrm{dom}(h) \mid t \in T\} = \{t' \circ h \mid t' \in T'\}$
$\iff (\forall t \in T : \exists t' \in T' : t \restriction \mathrm{dom}(h) = t' \circ h)$ en
$(\forall t' \in T' : \exists t \in T : t \restriction \mathrm{dom}(h) = t' \circ h)$
$\iff T \subseteq \mathrm{dom}(R)$ en $T' \subseteq \mathrm{rng}(R)$;

(2) volgt nu uit (0) en de voorgaande equivalenties.

(3) Stel $(t\,;\,t') \in R$ en $(t\,;\,t'') \in R$; we moeten nu bewijzen dat $t' = t''$.
Uit $(t\,;\,t') \in R$ en $(t\,;\,t'') \in R$ volgt dat $t' \in T'$ en $t'' \in T'$ en dat $t \upharpoonright \mathrm{dom}(h) = t' \circ h$ en $t \upharpoonright \mathrm{dom}(h) = t'' \circ h$;
dus $t' \circ h = t'' \circ h$; hieruit volgt dat $t' \upharpoonright \mathrm{rng}(h) = t'' \upharpoonright \mathrm{rng}(h)$;
uit het gegeven dat $\mathrm{rng}(h)$ u.i. in T' is volgt nu dat $t' = t''$.

(4) Stel $(t\,;\,t') \in R$ en $(s\,;\,t') \in R$; we moeten nu bewijzen dat $t = s$.
Uit $(t\,;\,t') \in R$ en $(s\,;\,t') \in R$ volgt dat $t \in T$ en $s \in T$ en dat $t \upharpoonright \mathrm{dom}(h) = t' \circ h$ en $s \upharpoonright \mathrm{dom}(h) = t' \circ h$;
dus $t \upharpoonright \mathrm{dom}(h) = s \upharpoonright \mathrm{dom}(h)$;
uit het gegeven dat $\mathrm{dom}(h)$ u.i. in T is volgt nu dat $t = s$.

□

OPGAVEN

2.36. Ga voor elk van de onderstaande beweringen over de tabellen T1 en T2 uit Voorbeeld 2.7 de juistheid na.

(a) {(ANR; AFDNR)} verbindt T2 met T1;

(b) {(AFDNR; ANR)} verbindt T1 met T2;

(c) {(MANNR; NR)} verbindt T2 met T1;

(d) {(NR; MANNR)} verbindt T1 met T2;

(e) $\mathrm{h4}^{-1}$ verbindt T1 met T2.

2.37. Beantwoord de volgende vragen over de attributentransformatie h4 uit Voorbeeld 2.7.

(a) Is dom(h4) (minimaal) u.i. in T2?

(b) Is rng(h4) (minimaal) u.i. in T1?

(c) Is de door h4 geïnduceerde associatie op T2 × T1 een <u>injectieve</u> <u>functie</u> <u>van</u> T2 <u>op</u> T1? (Controleer hierbij elk van de vier onderstreepte eigenschappen.)

2.38. Geldt voor elk tweetal tabellen T en T': $\varnothing$ verbindt T met T'?

2.39. Als T en T' verzamelingen functies zijn en h is een functie en h verbindt T met T' en $h' \subseteq h$, dan verbindt ook h' de verzameling T met T'. Bewijs dit.

2.40. Bewijs Lemma 2.15.

2.41. Geldt in Lemma 2.16, (3) en (4), ook "dan en slechts dan"?
(Geef een bewijs dan wel een tegenvoorbeeld.)

2.42. Herschrijf de definitie van VBU in Voorbeeld 1.3 met behulp van (bilaterale) verbinding.

□

2.2 BEGRIPPEN BETREFFENDE TABELLENVERZAMELINGEN

De begrippen uit §2.1.2 en §2.1.3, die zijn gedefinieerd voor *tabellen*, zullen we in §2.2.1 en §2.2.2 generaliseren tot een analogon op het niveau van tabellen*verzamelingen*. Om de verschillende niveaus goed te onderscheiden - hetgeen in de literatuur en in de praktijk vaak niet of nauwelijks gebeurt - zullen we voor overeenkomstige begrippen op verschillende niveaus telkens verschillende termen gebruiken.

In Voorbeeld 1.3 zagen we reeds twee tabellenverzamelingen, te weten de verzamelingen WM en WA. Voordat we de afzonderlijke begrippen betreffende tabellenverzamelingen introduceren, geven we eerst nog een voorbeeld van een tabellenverzameling.

VOORBEELD 2.8:

We definiëren de tabellenverzameling WP (betreffende personen) met behulp van een viertal hulpfuncties, in casu F1, F2, F3 en F4. De functies F1, F2 en F3 worden gebruikt in de definitie van de objectkarakterisering F4. Naast de definities zullen we ook weer enige informele toelichting geven.

Definities	*Toelichting*
F1 = {(STR ; Chs(50)),	straat
(HNR ; [1 .. 5000])};	huisnummer
F2 = {(NR ; [1000 .. 9999]),	nummer van een postcode
(LC ; Chs(2))};	letters van een postcode
F3 = {(DG ; [1 .. 31]),	dag
(MND; [1 .. 12]),	maand
(JR ; $\mathbb{N}$)};	jaar
F4 = {(NR ; $\mathbb{N} - \{0\}$),	identiteitsnummer
(NM ; Chs(40)),	naam
(ADR ; Π(F1)),	adres
(PC ; Π(F2)),	postcode
(WPL ; Chs(30)),	woonplaats
(GBD ; Π(F3)),	geboortedatum
(GPL ; Chs(30)),	geboorteplaats
(NMK; **P** (Chs(30)))};	namen der kinderen

$$
\begin{aligned}
\mathrm{WP} = \{T \subseteq \Pi(\mathrm{F4}) \mid\ & \{\mathrm{NR}\} \text{ is u.i. in } T \text{ en} \\
& \{\mathrm{PC}\} \rightarrow \{\mathrm{WPL}\} \text{ in } T \text{ en} \\
& \{\mathrm{ADR}, \mathrm{WPL}\} \rightarrow \{\mathrm{PC}\} \text{ in } T \text{ en} \\
& \forall t \in T\text{: als } t(\mathrm{NR}) \neq 1 \text{ dan } \exists t' \in T\text{: } t'(\mathrm{NR}) = t(\mathrm{NR}) - 1\}
\end{aligned}
$$

Figuur 2.7 representeert een element TP van WP.

NR	NM	ADR		PC		WPL	GBD			GPL	NMK
		STR	HNR	NR	LC		DG	MND	JR		
1	E.F. CODD	PARKLAAN	85	1234	AY	BEST	10	6	1962	LEEK	ø
2	C.J. DATE	SCHOOLWEG	34	9997	JL	SON	26	2	1988	SON	{JAN, PIET, EL
3	R. FAGIN	KERKSTR.	269	1501	TC	SON	16	1	1925	EDE	{AN, JAN}
4	M.E. SENKO	GR. MARKT	1	5665	AL	EDE	5	10	1953	VELP	{MARIE}

Figuur 2.7: De tabel TP, een element van WP

□ Voorbeeld 2.8.

We hebben in Voorbeeld 2.8 tevens laten zien dat attributen zelf weer "subattributen" kunnen hebben, namelijk als de attribuutwaarden functies zijn. Voorbeelden zijn de attributen ADR, PC en GBD met als verzamelingen subattributen respectievelijk {STR, HNR}, {NR, LC} en {DG, MND, JR}. In principe kunnen we zo'n "nesting" van attributen natuurlijk willekeurig diep maken. Of de huidige database-managementsystemen dergelijke constructies ook aankunnen is echter een heel andere vraag.

2.2.1 Permanente afhankelijkheid

We noemen het analogon van momentane afhankelijkheid op het niveau van tabellenverzamelingen *permanente afhankelijkheid*. In het bijzonder noemen we *C permanent afhankelijk van B in W* als *C* momentaan afhankelijk is van *B* in elk element van *W*. We noemen permanente afhankelijkheid ook wel *structurele afhankelijkheid* (en momentane afhankelijkheid ook wel *incidentele afhankelijkheid*). Ter onderscheiding van de notatie "$B \rightarrow C$ in T" voor momentane afhankelijkheid in een tabel T gebruiken we voor het gegeneraliseerde begrip, permanente afhankelijkheid in een tabellenverzameling W, de notatie "$B \stackrel{\frown}{\rightarrow} C$ in W".

DEFINITIE 2.9:

Als A, B en C verzamelingen zijn en W is een verzameling tabellen over A, dan:
$B \hat{\rightarrow} C$ **in** $W \overset{D}{\Longleftrightarrow} \forall T \in W : B \rightarrow C$ in T.

Evenals bij Definitie 2.4 zijn ook hier de interessante gevallen weer die waarin $B \subseteq A$ en $C \subseteq A$.

In de database-literatuur wordt vaak de term *functionele afhankelijkheid* gebruikt; op vele plaatsen is het echter niet duidelijk of daarmee momentane afhankelijkheid dan wel permanente afhankelijkheid wordt bedoeld.

VOORBEELD 2.9:

Met de tabellenverzameling WM van Voorbeeld 1.3 in gedachten laten we de in Voorbeeld 2.5 genoemde momentane afhankelijkheden in de tabel T1 eens de revue passeren. We kunnen dan uit de definitie van WM concluderen dat de in Voorbeeld 2.5(a) genoemde *momentane* afhankelijkheid in T1 ook een *permanente* afhankelijkheid in WM is:

(a) $\{NR\} \hat{\rightarrow} \{NR, NAAM, SAL, GESL, AFDNR\}$ in WM.

De afhankelijkheden genoemd in (g) en (h) van Voorbeeld 2.5 bleken al niet te gelden in T1, en dus zeker niet in WM (want T1 $\in$ WM). Ook voor de andere afhankelijkheden kunnen we echter tegenvoorbeelden geven. Zij bijvoorbeeld T12 de tabel die we uit de tabel T1 verkrijgen door van het tupel met medewerkersnummer 8 het afdelingsnummer te veranderen van 1 in 2; zie Figuur 2.8. Het is dan eenvoudig na te gaan dat T12 $\in$ WM. Echter, de afhankelijkheden in (c), (d) en (f) van Voorbeeld 2.5 gelden niet in T12, en daarmee dus ook niet in WM. De gevallen (b) en (e) worden in Opgave 2.44 behandeld.

NR	NAAM	SAL	GESL	AFDNR
7	JANSSEN	3109	V	2
8	JANSSEN	2200	M	2
9	DEKKER	2995	M	1

Figuur 2.8: De tabel T12

□ Voorbeeld 2.9.

OPGAVEN

2.43. Deze opgave gaat nader in op Voorbeeld 2.8 aan het begin van §2.2.

(a) Geef de eis in de laatste regel van de definitie van WP in woorden weer.

(b) Heeft WP oneindig veel elementen?

(c) Heeft WP oneindig grote elementen?

(d) Heeft WP eindige elementen van elke grootte?

Zij F40 de objectkarakterisering die we verkrijgen door in de definitie van F4 in Voorbeeld 2.8 de verzameling $\mathbb{N} - \{0\}$ te vervangen door $\mathbb{N}$ en zij WP0 de tabellenverzameling die we verkrijgen door in de definitie van WP het gegeneraliseerde produkt Π(F4) te vervangen door Π(F40).

(e) Ga na of Π(F4) bevat is in Π(F40), en omgekeerd.

(f) Ga na of WP bevat is in WP0, en omgekeerd.

Zij nu FI de objectkarakterisering die we verkrijgen door in de definitie van F40 de verzameling $\mathbb{N}$ te vervangen door $\mathbb{Z}$ en WI de tabellenverzameling die we verkrijgen door in de definitie van WP0 het gegeneraliseerde produkt Π(F40) te vervangen door Π(FI).

(g) Ga na of Π(F40) bevat is in Π(FI), en omgekeerd.

(h) Ga na of WP0 bevat is in WI, en omgekeerd.

2.44. Toon aan dat de momentane afhankelijkheden genoemd in (b) en (e) van Voorbeeld 2.5 geen permanente afhankelijkheden in WM zijn.

2.45. Met de eigenschappen van momentane afhankelijkheid genoemd in de lemma's 2.7 tot en met 2.10 corresponderen overeenkomstige eigenschappen van permanente afhankelijkheid. Formuleer (en bewijs) die eigenschappen.

2.46. (a) Ga in Voorbeeld 2.8 na dat $\{NR\} \stackrel{\wedge}{\rightarrow} \{NR, ADR, WPL\}$ in WP, $\{ADR, WPL\} \stackrel{\wedge}{\rightarrow} \{PC\}$ in WP en $\{ADR, PC\} \stackrel{\wedge}{\rightarrow} \{WPL\}$ in WP.

(b) Is $\{NR\}$ permanent afhankelijk van $\{NM, ADR, WPL\}$ in WP?

2.47. Figuur 2.9 representeert twee tabellen; zij T6 de tabel gerepresenteerd in (a) en T7 de tabel gerepresenteerd in (b). We definiëren W1 = {T1, T6, T7} en W2 = W1 ∪ {∅}, waarbij T1 de tabel uit Voorbeeld 2.5 is. Bepaal alle permanente afhankelijkheden in W1 respectievelijk W2 die van de vorm $B \hat{\rightarrow} \{c\}$ zijn, waarbij $c \in$ {NR, NAAM, SAL, GESL, AFDNR} – B en B minimaal is in de volgende zin:
$\forall B' \subseteq B$: als $B' \hat{\rightarrow} \{c\}$ in W1 (respectievelijk W2) dan $B' = B$.

NR	NAAM	SAL	GESL	AFDNR
8	JANSSEN	2451	M	1
2	HEKKING	3109	M	2
3	HEKKING	2451	M	3
9	DEKKER	3235	M	4

(a) De tabel T6

NR	NAAM	SAL	GESL	AFDNR
8	JANSSEN	2452	M	1
1	JANSEN	2452	V	1

(b) De tabel T7

Figuur 2.9

2.48. Zij WP1 de verzameling van alle elementen T van de in Voorbeeld 2.8 gedefinieerde tabellenverzameling WP die voldoen aan de eis dat elk tweetal tupels in T met dezelfde postcode en hetzelfde huisnummer ook hetzelfde adres heeft.

(a) Geef een formele definitie van WP1.

(b) Heeft de zojuist omschreven eis de vorm van een permanente afhankelijkheid in WP1?

(c) Is WP1 een echte deelverzameling van WP?
Licht uw antwoord toe.

□

2.2.2 Sleutels en minimale sleutels

We noemen B een *sleutel* van een tabellenverzameling W als B uniek identificerend is in élk element van W. Als bovendien geen enkele echte deelverzameling van B een sleutel van W is dan noemen we B ook wel een *minimale sleutel* van W. Om precies te zijn:

DEFINITIE 2.10:
Als A een verzameling is en W is een verzameling tabellen over A, dan:
B is een **sleutel van** W $\overset{D}{\Longleftrightarrow}$ B is een verzameling en
$\forall T \in W$: B is u.i. in T.

DEFINITIE 2.11:
Als A een verzameling is en W is een verzameling tabellen over A, dan:
B is een **minimale sleutel van** W $\overset{D}{\Longleftrightarrow}$ B is een sleutel van W en
$\forall B' \subset B$: B' is geen sleutel van W.

We willen er op wijzen dat in de (Nederlandstalige) literatuur de term *sleutel* ook wel wordt gebruikt in de zin van Definitie 2.11. In de Engelstalige literatuur wordt de overeenkomstige term *key* ook in beide betekenissen gebruikt, terwijl de term *superkey* uitsluitend in de zin van Definitie 2.10 voorkomt.

VOORBEELD 2.10:

De verzamelingen {NR, NAAM} en {NR} zijn voorbeelden van sleutels van de tabellenverzameling WM uit Voorbeeld 1.3; hieruit volgt overigens meteen dat {NR, NAAM] geen *minimale* sleutel van WM is. Omdat ∅ geen sleutel van WM is, is de sleutel {NR} van WM wél minimaal.

De tabellenverzameling WA uit Voorbeeld 1.3 heeft precies twee minimale sleutels, namelijk {ANR} en {NAAM}.

☐ Voorbeeld 2.10.

Voor tabellen*verzamelingen* is Lemma 2.17 het analogon van Lemma 2.12.

LEMMA 2.17:
Als A een verzameling is en W is een verzameling tabellen over A en $B \subseteq A$ en $C \subseteq A$ en $D \subseteq A$, dan:

(a) B is een sleutel van W desda $B \hat{\rightarrow} A$ in W;

(b) B is een sleutel van W desda $\forall a \in A - B : B \hat{\rightarrow} \{a\}$ in W;

(c) A is een sleutel van W;

(d) als B een sleutel van W is en $B \subseteq C$, dan is C ook een sleutel van W;

(e) als B een sleutel van W is en $C \hat{\rightarrow} B$ in W, dan is C ook een sleutel van W;

(f) als $B \hat{\rightarrow} C$ in W en $D \hat{\rightarrow} (A - C)$ in W, dan is $B \cup D$ een sleutel van W;

(g) $\varnothing$ is een sleutel van W desda $\forall T \in W : | T | \leq 1$.

We besluiten §2.2.2 met de invoering van enige additionele begrippen en notaties aangaande minimale sleutels van tabellenverzamelingen. Laat W een verzameling tabellen over een verzameling A zijn. We beginnen met de opmerking dat, op een paar uitzonderlijke gevallen van W na, de verzameling A is uit te drukken in termen van W: $A = \mathrm{He}(\bigcup W)$; zie Opgave 2.56. We zullen $\mathrm{He}(\bigcup W)$ daarom *de heading van W* noemen en aanduiden met Head(W). Verder duiden we de verzameling van alle minimale sleutels van W aan met Ms(W):

DEFINITIE 2.12:
Als A een verzameling is en W is een verzameling tabellen over A, dan:

(a) $\mathrm{Head}(W) \stackrel{D}{=} \mathrm{He}(\bigcup W)$;

(b) $\mathrm{Ms}(W) \stackrel{D}{=} \{B \subseteq \mathrm{Head}(W) \mid B \text{ is een minimale sleutel van } W\}$.

Een element van Head(W) noemen we een *attribuut van W*. Een element van een minimale sleutel van W noemen we wel een *primair* attribuut van W; de andere attributen van W noemen we *secundair*. De verzameling van alle primaire attributen van W duiden we aan met Prim(W) en de verzameling van alle secundaire attributen van W met Sec(W):

DEFINITIE 2.13:
Als A een verzameling is en W is een verzameling tabellen over A, dan:

(a) $\mathrm{Prim}(W) \stackrel{D}{=} \bigcup \mathrm{Ms}(W)$;

(b) $\mathrm{Sec}(W) \stackrel{D}{=} \mathrm{Head}(W) - \mathrm{Prim}(W)$.

VOORBEELD 2.11:

In Opgave 2.49 wordt gevraagd te bewijzen dat

$$\mathrm{Ms}(\mathrm{WA}) = \{\{\mathrm{ANR}\}, \{\mathrm{NAAM}\}\} \text{ en } \mathrm{Ms}(\mathrm{WM}) = \{\{\mathrm{NR}\}\}.$$

Met behulp van Definitie 2.13 (en Lemma 0.1) kunnen we hieruit vervolgens de verzamelingen primaire attributen van WM en WA berekenen:

$$\begin{aligned}
\mathrm{Prim}(\mathrm{WM}) &= \bigcup \mathrm{Ms}(\mathrm{WM}) = \bigcup \{\{\mathrm{NR}\}\} = \{\mathrm{NR}\} \text{ en} \\
\mathrm{Prim}(\mathrm{WA}) &= \bigcup \mathrm{Ms}(\mathrm{WA}) = \bigcup \{\{\mathrm{ANR}\}, \{\mathrm{NAAM}\}\} \\
&= \{\mathrm{ANR}\} \cup \{\mathrm{NAAM}\} = \{\mathrm{ANR}, \mathrm{NAAM}\}.
\end{aligned}$$

□ Voorbeeld 2.11.

OPGAVEN

2.49. Bewijs dat de attributenverzamelingen {ANR} en {NAAM} van WA en {NR} van WM de enige minimale sleutels van die tabellenverzamelingen uit Voorbeeld 1.3 zijn.

2.50. Bewijs Lemma 2.17.

2.51. Formuleer en bewijs een analogon van Lemma 2.13 voor tabellenverzamelingen.

2.52. (a) Bepaal de verzameling van alle minimale sleutels van WP uit Voorbeeld 2.8.
(b) Bepaal de verzameling van alle minimale sleutels van W1 uit Opgave 2.47.

2.53. Als A een verzameling is en W is een verzameling tabellen over A en $B' \subseteq A$ en $B \subseteq B'$ en B is een minimale sleutel van W en $W' \subseteq W$ en h is een injectieve functie op B, is dan:

(a) B een minimale sleutel van W'?

(b) B een minimale sleutel van $\{T \upharpoonright B' \mid T \in W\}$?

(c) $\mathrm{dom}(h)$ een minimale sleutel van $\{T \infty h \mid T \in W\}$?

Geef telkens een bewijs of een tegenvoorbeeld.

2.54. Bewijs dat elke eindige sleutel van een tabellenverzameling W een minimale sleutel van W omvat.

2.55. Toon met behulp van een voorbeeld aan dat niet elke (oneindige) sleutel een minimale sleutel omvat.

2.56. (a) Voor welke tabellenverzamelingen W over een verzameling A is A eenduidig door W bepaald?

(b) Bewijs dat in dat geval $A = \mathrm{He}(\bigcup W)$.

(c) Bereken $\mathrm{He}(\bigcup W)$ in de andere gevallen.

□

2.2.3 Enige normaalvormen

Voor tabellenverzamelingen worden er in de literatuur diverse zogenaamde "normaalvormen" onderscheiden. We zullen in deze paragraaf alleen de twee belangrijkste normaalvormen behandelen, namelijk de *Boyce-Codd normaalvorm* (afkorting: BCNF) en de *derde normaalvorm* (afkorting: 3NF).

We definiëren eerst wanneer een tabellenverzameling in Boyce-Codd normaalvorm is:

DEFINITIE 2.14:
Als A een verzameling is en W is een verzameling tabellen over A, dan:
W is in BCNF $\overset{D}{\Longleftrightarrow} \forall B \subseteq \mathrm{Head}(W) : \forall a \in \mathrm{Head}(W)$: als $B \hat{\rightarrow} \{a\}$ in W en $a \notin B$ dan is B een sleutel van W.

In de literatuur wordt voor BCNF en voor 3NF ook nog geëist dat de tabellenverzameling in 1NF (*eerste normaalvorm*) is, dat wil zeggen dat elke attribuutwaarde "atomair" is. Het begrip "atomair" wordt dan meestal omschreven als "nondecomposable by the underlying database management system", zie bijvoorbeeld [Ya 86]. (Als kritische kanttekening merken we hierbij op dat een DBMS bijvoorbeeld meestal in staat is om strings te decomponeren in substrings . . . Is het DBMS nu te krachtig of zijn strings niet als attribuutwaarden toegestaan?)

We laten de (nogal vage) 1NF-eis bij al onze normaalvormen weg, niet alleen bij BCNF en 3NF maar later ook bij de definitie van 2NF en 4NF (in de herhalingsopgaven aan het eind van Hoofdstuk 2). De reden daarvoor is niet alleen dat deze eis in feite niet goed

formaliseerbaar is, maar vooral ook dat deze (DBMS-afhankelijke) eis naar onze mening niet op dit niveau thuishoort. Daar komt nog bij dat "non-1NF databases" in diverse toepassingen vaak wenselijk zijn. Dit besef schijnt ook elders door te dringen, zie bijvoorbeeld [RKS 88].

De oorspronkelijke definitie van 3NF is te vinden in [Co 72] en van BCNF in [Co 74].

We geven nu onze definitie van 3NF:

DEFINITIE 2.15:
Als A een verzameling is en W is een verzameling tabellen over A, dan:
W **is in 3NF** $\overset{D}{\Longleftrightarrow} \forall B \subseteq \text{Head}(W) : \forall a \in \text{Sec}(W)$: als $B \hat{\to} \{a\}$ in W en a $\notin B$
dan is B een sleutel van W.

Informeel kunnen we de twee normaalvormen als volgt onder woorden brengen: W is in BCNF als elke attributenverzameling B die een attribuut buiten B "bepaalt" een sleutel is, en W is in 3NF als elke attributenverzameling B die een *secundair* attribuut buiten B "bepaalt" een sleutel is. BCNF is dus een sterkere eigenschap dan 3NF:

LEMMA 2.18:
Als A een verzameling is en W is een verzameling tabellen over A, dan:
Als W in BCNF is, dan is W ook in 3NF.

VOORBEELD 2.12:

We geven een voorbeeld van een tabellenverzameling die wél in 3NF maar niet in BCNF is. We laten ons daartoe inspireren door de structuur van het postcodeboek. (We nemen hierbij uit Voorbeeld 2.8 de hulpfunctie F2 over.)

F2 =	{(NR ; [1000 .. 9999]),	nummer van de postcode
	(LC ; Chs (2))};	letters van de postcode
FPCB =	{(STR ; Chs(50)),	straat
	(HNR; Chs(6)),	huisnummer
	(WPL; Chs(40)),	woonplaats
	(PC ; Π(F2))};	postcode

WPCB = $\{T \mid T \subseteq \Pi(\text{FPCB})$ en
{STR, HNR, WPL} is u.i. in T en
{PC, HNR} is u.i. in T en
$\{PC\} \to \{WPL\}$ in $T\}$

We merken op dat Ms(WPCB) = {{STR, HNR, WPL}, {PC, HNR}}, zodat dus alle attributen van WPCB primaire attributen zijn. Hieruit volgt dan op triviale wijze dat WPCB in 3NF is. Dat WPCB niet in BCNF is volgt uit het feit dat {PC} $\hat{\rightarrow}$ {WPL} in WPCB, terwijl {PC} geen sleutel van WPCB is.

□ Voorbeeld 2.12.

OPGAVEN

2.57. Ga voor de volgende tabellenverzamelingen na of deze in BCNF zijn.

(a) WP uit Voorbeeld 2.8;

(b) WP1 uit Opgave 2.48;

(c) W1 uit Opgave 2.47;

(d) W2 uit Opgave 2.47;

(e) $\{\varnothing\}$;

(f) $\varnothing$.

2.58. Als W een tabellenverzameling over een verzameling A is en $C \subseteq A$ en $W' \subseteq W$ en W is in BCNF, zijn dan ook W' en $\{T \restriction C \mid T \in W\}$ in BCNF?
Als W' in BCNF is, is dan ook W in BCNF?

2.59. Bewijs dat Ms(WPCB) in Voorbeeld 2.12 inderdaad gelijk is aan {{STR, HNR, WPL}, {PC, HNR}}.

2.60. Geef het bewijs van Lemma 2.18.

2.61. Ga voor de tabellenverzamelingen uit Opgave 2.57 na of deze in 3NF zijn.

2.62. Als W een tabellenverzameling over een verzameling A is en $C \subseteq A$ en $W' \subseteq W$ en W is in 3NF, zijn dan ook W' en $\{T \restriction C \mid T \in W\}$ in 3NF?
Als W' in 3NF is, is dan ook W in 3NF?

□

2.3 BEGRIPPEN BETREFFENDE DATABASE-UNIVERSA

In §2.3.1 beginnen we met de behandeling van de eigenschappen van enige operaties op DB-universa.

De in §2.2 ingevoerde begrippen *permanente afhankelijkheid* en *(minimale) sleutel* willen we ook gebruiken op het niveau van DB-universa. De precieze terminologie en de bijbehorende notaties worden geïntroduceerd in §2.3.2.

In §2.3.3 zullen we de in §2.2.3 gedefinieerde normaalvormen voor tabellenverzamelingen generaliseren tot normaalvormen voor DB-universa. In feite gaat het ons namelijk niet zozeer om normaalvormen voor een "losse" tabellenverzameling maar meer om normaalvormen voor een DB-universum als geheel.

In §2.3.4 openen we met een generalisatie van het in §2.1.4 ingevoerde verbindingsbegrip. Verder definiëren we in deze paragraaf het conceptueel belangrijke begrip *database-functie*.

Tot nu toe hebben we slechts één voorbeeld van een DB-universum laten zien, te weten het DB-universum VBU in Voorbeeld 1.3. Voordat we de afzonderlijke begrippen betreffende DB-universa behandelen, willen we daarom eerst nóg een voorbeeld van een DB-universum geven.

VOORBEELD 2.13:

We herhalen de "modules" F1, F2, F3, F4 en WP uit Voorbeeld 2.8 en breiden dit voorbeeld uit tot een DB-universum U1 (betreffende personen en gemeenten in Nederland). We doen dit door toevoeging van een deelverzameling S2 van Chs(2), een objectkarakterisering F5, een tabellenverzameling WG, een database-karakterisering F6 en tenslotte het DB-universum U1. (We merken overigens op dat deze structuur strict genomen niet precies de Nederlandse situatie weerspiegelt, onder andere omdat we (eenvoudigheidshalve) woonplaatsen en gemeenten met elkaar hebben geïdentificeerd. Ook hoeft een gemeentenaam niet uniek te zijn. In Opgave 2.105 komen we hierop terug.)

$$\begin{aligned} F1 = \{&(\text{STR} \; ; \text{Chs}(50)), \\ &(\text{HNR} \; ; [1 \,..\, 5000])\}; \end{aligned}$$

$$\begin{aligned} F2 = \{&(\text{NR} \; ; [1000 \,..\, 9999]), \\ &(\text{LC} \; ; \text{Chs}(2))\}; \end{aligned}$$

$$\begin{aligned} F3 = \{&(\text{DG} \; ; [1 \,..\, 31]), \\ &(\text{MND} ; [1 \,..\, 12]), \\ &(\text{JR} \; ; \mathbb{N})\}; \end{aligned}$$

$$\begin{aligned} F4 = \{&(\text{NR} \; ; \mathbb{N} - \{0\}), \\ &(\text{NM} \; ; \text{Chs}(40)), \\ &(\text{ADR} \; ; \Pi(\text{F1})), \\ &(\text{PC} \; ; \Pi(\text{F2})), \\ &(\text{WPL} \; ; \text{Chs}(30)), \\ &(\text{GBD} \; ; \Pi(\text{F3})), \\ &(\text{GPL} \; ; \text{Chs}(30)), \\ &(\text{NMK} ; \mathbf{P}(\text{Chs}(30)))\}; \end{aligned}$$

$$\begin{aligned} WP = \{T \subseteq \Pi(\text{F4}) \mid\ & \{\text{NR}\} \text{ is u.i. in } T \text{ en} \\ & \{\text{PC}\} \rightarrow \{\text{WPL}\} \text{ in } T \text{ en} \\ & \{\text{ADR, WPL}\} \rightarrow \{\text{PC}\} \text{ in } T \text{ en} \\ & \forall t \in T\text{: als } t(\text{NR}) \neq 1 \text{ dan } \exists t' \in T : t'(\text{NR}) = t(\text{NR}) - 1\}; \end{aligned}$$

$$\begin{aligned} S2 = \{&\text{'GR'}, \text{'FR'}, \text{'DR'}, \text{'OV'}, \\ &\text{'FL'}, \text{'GE'}, \text{'UT'}, \text{'NH'}, \\ &\text{'ZH'}, \text{'ZE'}, \text{'NB'}, \text{'LI'}\}; \end{aligned}$$

de provincies van Nederland

$F5 = \{(\text{WPL} \; ; \text{Chs}(30)),$	gemeentenaam
$(\text{PROV} ; \text{S2}),$	provincie
$(\text{INWA} ; \mathbb{N}),$	inwoneraantal
$(\text{OPVL} ; \mathbb{N}),$	oppervlakte (in hectare)
$(\text{NR} \; ; \mathbb{N})\};$	identiteitsnummer van de burgemeester

$$WG = \{T \subseteq \Pi(\text{F5}) \mid \{\text{WPL}\} \text{ is u.i. in } T\};$$

$$\begin{aligned} F6 = \{&(\text{PERS} ; \text{WP}), \\ &(\text{GEM} \; ; \text{WG})\}; \end{aligned}$$

$$\begin{aligned} U1 = \{v \in \Pi(\text{F6}) \mid\ & \text{id}(\{\text{WPL}\}) \text{ verbindt } v(\text{PERS}) \text{ met } v(\text{GEM}) \text{ en} && \text{(a)} \\ & \text{id}(\{\text{NR, WPL}\}) \text{ verbindt } v(\text{GEM}) \text{ met } v(\text{PERS})\} && \text{(b)} \end{aligned}$$

☐ Voorbeeld 2.13.

2.3.1 Operaties op database-universa

In deze paragraaf zijn we geïnteresseerd in operaties die aan een of meer DB-universa weer een DB-universum toevoegen.

Omdat een DB-universum een verzameling is, zijn begrippen als deelverzameling, vereniging, doorsnede en verschil ook van toepassing op DB-universa. Dergelijke operaties leveren meestal ook weer DB-universa op, en wel over het oorspronkelijke DB-skelet, zie (1), (2) en (3) van Stelling 2.1 voor de details.

We roepen verder ook nog Stelling 1.1 in herinnering die zegt dat elk DB-universum over een verzamelingsfunctie g tevens een tabel over dom(g) is. Dat betekent dus dat alle begrippen, notaties en operaties die we intussen voor tabellen hebben gedefinieerd eveneens op DB-universa van toepassing zijn!

In het bijzonder kunnen we dus de in §2.1.1 ingevoerde tabeloperaties ook op DB-universa toepassen; die operaties leveren volgens de resultaten in die paragraaf telkens weer tabellen op. De vraag is echter of het toepassen van zo'n operatie op DB-universa ook weer een *DB-universum* oplevert, en zo ja, over welk DB-skelet. Vanaf onderdeel (4) gaat Stelling 2.1 ook op deze vraag in en levert daarbij een aantal interessante resultaten op. We zullen echter eerst kort omschrijven waar die operaties uit §2.1.1 voor DB-universa op neerkomen:

– Projectie (op een verzameling tabelindexen) correspondeert met het beperken van het DB-universum (en van de DB-toestanden) tot een "module" of "subschema" van (al dan niet "bij elkaar behorende") tabelindexen.
– De join van twee DB-universa met compatibele DB-skeletten correspondeert met het "integreren" van alle mogelijke paren DB-toestanden die voor gelijknamige tabelindexen dezelfde (tabel)waarde hebben, tot een DB-toestand die alle tabelindexen van de oorspronkelijke twee DB-toestanden bevat.
– Herbenoeming, tenslotte, komt bij DB-universa neer op het herbenoemen van *tabelindexen.*

Met de nummering binnen Stelling 2.1 volgen we (tot en met (6)) de nummering van de lemma's in §2.1.1 over diezelfde operaties.

STELLING 2.1:
Als g en g' verzamelingsfuncties zijn
en U en V zijn DB-universa over g
en U' is een DB-universum over g'
en $D \subseteq U$ en $X \subseteq \text{dom}(g)$ en h is een functie, dan:

(1)	D	is een DB-universum over g;
(2)	$U \cap V$	is een DB-universum over g en
	$U - V$	is een DB-universum over g;
(3)	$U \cup V$	is een DB-universum over g;
(4)	$U \upharpoonright X$	is een DB-universum over $g \upharpoonright X$;
(5)	$U \bowtie U'$	is een DB-universum over $g \cup g'$ mits g en g' compatibel zijn;
(6)	$U \infty h$	is een DB-universum over $g \circ h$;
(7)	$U \not\upharpoonright X$	is een DB-universum over $g \not\upharpoonright X$.

Bewijs:

(1) Elk element van U is een DB-toestand over g, dus elk element van D is dat ook.

(2) Dit volgt direct uit (1).

(3) Elk element van U is een DB-toestand over g en
elk element van V is een DB-toestand over g,
dus elk element van $U \cup V$ is dat ook.

(4) In dit onderdeel maken we (stilzwijgend) intensief gebruik van Lemma 0.11:
Elk element van $U \upharpoonright X$ is van de vorm $v \upharpoonright X$, waarbij $v \in U$.
Nu geldt dat $\text{dom}(v \upharpoonright X) = X = \text{dom}(g \upharpoonright X)$, dus $v \upharpoonright X$ is een functie over $\text{dom}(g \upharpoonright X)$.
Verder geldt voor elke $E \in \text{dom}(g \upharpoonright X)$ dat $(v \upharpoonright X)(E) = v(E)$,
dus $(v \upharpoonright X)(E)$ is een tabel over $g(E)$, ofwel over $(g \upharpoonright X)(E)$.
Volgens Definitie 1.2 is $v \upharpoonright X$ dus een DB-toestand over $g \upharpoonright X$.
Volgens Definitie 1.3 is $U \upharpoonright X$ dus een DB-universum over $g \upharpoonright X$.

(5) In dit onderdeel maken we gebruik van Lemma 0.2(b):
Elk element van $U \bowtie U'$ is van de vorm $v \cup v'$,
waarbij $v \in U$ en $v' \in U'$ en $v \cup v'$ een functie is.
Nu geldt dat $\text{dom}(v \cup v') = \text{dom}(v) \cup \text{dom}(v') = \text{dom}(g) \cup \text{dom}(g') = \text{dom}(g \cup g')$,
dus $v \cup v'$ is een functie over $\text{dom}(g \cup g')$.
Verder geldt voor elke $E \in \text{dom}(g \cup g')$ dat

– $E \in \text{dom}(v)$, zodat $(v \cup v')(E) = v(E)$, een tabel over $g(E)$, of

– $E \in \text{dom}(v')$, zodat $(v \cup v')(E) = v'(E)$, een tabel over $g'(E)$.

Omdat g en g' compatibel zijn, is dus $(v \cup v')(E)$ in beide gevallen een tabel over $(g \cup g')(E)$.

Volgens Definitie 1.2 is $v \cup v'$ dus een DB-toestand over $g \cup g'$.

Volgens Definitie 1.3 is $U \bowtie U'$ dus een DB-universum over $g \cup g'$.

(6) Elk element van $U \infty h$ is van de vorm $v \circ h$, waarbij $v \in U$.
Omdat $\text{dom}(v) = \text{dom}(g)$, geldt volgens Lemma 0.9(g) dat $\text{dom}(v \circ h) = \text{dom}(g \circ h)$; dus $v \circ h$ is een functie over $\text{dom}(g \circ h)$.
Verder geldt voor elke $E \in \text{dom}(g \circ h)$ dat $(v \circ h)(E) = v(h(E))$, dit is een tabel over $g(h(E))$, ofwel over $(g \circ h)(E)$.
Volgens Definitie 1.2 is $v \circ h$ dus een DB-toestand over $g \circ h$.
Volgens Definitie 1.3 is $U \infty h$ dus een DB-universum over $g \circ h$.

(7) Het bewijs van dit onderdeel is analoog aan dat van (4).
(Voor de definitie van $U \not\upharpoonright X$ verwijzen we naar Opgave 2.22.)

□

In de opgaven behorende bij §2.1.1 worden nog diverse eigenschappen opgesomd van de in Stelling 2.1 behandelde operaties bij toepassing op tabellen. Deze eigenschappen gelden dus evenzeer bij toepassing op DB-universa!

OPGAVEN

2.63. Schrijf het DB-skelet uit van het DB-universum U1 dat in Voorbeeld 2.13 is gedefinieerd.

2.64. Geef een informele omschrijving van de formele eisen (a) en (b) in de definitie van U1.

2.65. Geef een element van U1, bijvoorbeeld met behulp van de tabel TP in Figuur 2.7.

2.66. Is in elke DB-toestand conform U1 het inwoneraantal van een gemeente in overeenstemming met het aantal personen met die gemeente als woonplaats?
Zo ja, toon dit dan aan; zo nee, geef dan een formele definitie van het DB-universum U2 bestaande uit alle elementen van U1 die wél die eigenschap hebben.

2.67. Ga na in hoeverre de lemma's 2.4, 2.5 en 2.6 nog aangescherpt of verfijnd kunnen worden voor DB-universa. Licht uw bevindingen toe.

□

2.3.2 Permanente afhankelijkheid en (minimale) sleutels

We definiëren de begrippen permanente afhankelijkheid, sleutel en minimale sleutel met betrekking tot een tabelindex E van een DB-universum U door de overeenkomstige begrippen uit §2.2 toe te passen op de tabellenverzameling $\{v(E) \mid v \in U\}$. We zullen deze tabellenverzameling *het universum van E in U* noemen en aanduiden met $\mathrm{Un}(E,U)$.

DEFINITIE 2.16:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{He}(U)$, dan:
$\mathrm{Un}(E,U) \stackrel{D}{=} \{v(E) \mid v \in U\}$.

DEFINITIE 2.17:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{He}(U)$ en B en C zijn verzamelingen, dan:
$B \hat{\rightarrow} C$ **in** E **van** $U \stackrel{D}{\Longleftrightarrow} B \hat{\rightarrow} C$ in $\mathrm{Un}(E,U)$.

DEFINITIE 2.18:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{He}(U)$, dan:
(a) B is een **sleutel van** E **in** $U \stackrel{D}{\Longleftrightarrow} B$ is een sleutel van $\mathrm{Un}(E,U)$;
(b) B is een **minimale sleutel van** E **in** $U \stackrel{D}{\Longleftrightarrow} B$ is een minimale sleutel van $\mathrm{Un}(E,U)$.

Op analoge wijze tillen we de in de Definities 2.12 en 2.13 ingevoerde notaties betreffende een tabellenverzameling W naar het niveau van een tabelindex E van een DB-universum U. De nieuwe notaties zijn te onderscheiden van de gelijknamige oude notaties doordat ze twee parameters hebben in plaats van één.

DEFINITIE 2.19:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{He}(U)$, dan:

(a) $\mathrm{Head}(E,U) \stackrel{D}{=} \mathrm{Head}(\mathrm{Un}(E,U))$;

(b) $\mathrm{Ms}(E,U) \stackrel{D}{=} \mathrm{Ms}(\mathrm{Un}(E,U))$;

(c) $\mathrm{Prim}(E,U) \stackrel{D}{=} \mathrm{Prim}(\mathrm{Un}(E,U))$;

(d) $\mathrm{Sec}(E,U) \stackrel{D}{=} \mathrm{Sec}(\mathrm{Un}(E,U))$.

We zullen de voorgaande begrippen toelichten aan de hand van het in Voorbeeld 2.13 gedefinieerde DB-universum U1.

VOORBEELD 2.14:

We zullen eerst de verzamelingen Un(PERS, U1) en Un(GEM, U1), de universa van de tabelindexen PERS en GEM in U1, proberen te bepalen. Uit de definities van U1 en F6 volgt meteen dat

Un(PERS, U1) $\subseteq$ WP en Un(GEM, U1) $\subseteq$ WG.

Of deze inclusies ook gelijkheden zijn, is echter minder eenvoudig in te zien. We tonen eerst aan dat elke $T \in$ WP inderdaad kan optreden als PERS-tabel in een geschikt gekozen $v_T \in$ U1. We definiëren daartoe de functie v_T over {PERS, GEM} als volgt:

v_T (PERS) = T en

$$\begin{aligned} v_T\,(\mathrm{GEM}) = \{\{&(\mathrm{WPL}\ ;\ t(\mathrm{WPL})),\\ &(\mathrm{PROV};\ \text{'GR'}),\\ &(\mathrm{INWA};\ 20000),\\ &(\mathrm{OPVL};\ 6000),\\ &(\mathrm{NR}\ ;\ t(\mathrm{NR}))\}\\ &\mid t \in T \text{ en } \forall t' \in T\text{: als } t(\mathrm{WPL}) = t'(\mathrm{WPL}) \text{ dan } t(\mathrm{NR}) \leq t'(\mathrm{NR})\} \end{aligned} \qquad \text{(c)}$$

De functie v_T blijkt een element van U1 te zijn (zie Opgave 2.68(b)). Hiermee hebben we dan dus aangetoond dat WP $\subseteq$ Un(PERS, U1), en derhalve dat Un(PERS, U1) in feite gelijk is aan WP. (Dat zo'n DB-toestand v_T misschien niet erg realistisch aandoet, is natuurlijk in dit verband niet relevant.)

We laten nu zien dat het universum van GEM in U1 een *echte* deelverzameling van WG is. We zullen zelfs iets meer bewijzen, namelijk dat

Un(GEM, U1) $\subseteq$ {$T \in$ WG | {NR} is u.i. in T}. (0)

Gemakshalve zullen we {$T \in$ WG | {NR} is u.i. in T} in het vervolg WG2 noemen. Aangezien WG2 een echte deelverzameling van WG is (zie ook Opgave 2.69(a)), is Un(GEM, U1) dat dus ook. (In feite blijkt Un(GEM, U1) zelfs ook een echte deelverzameling van WG2 te zijn, zie Opgave 2.69(b). Sterker nog, het is zelfs vrij moeilijk om Un(GEM, U1) expliciet als deelverzameling van WG te karakteriseren, zie Opgave 2.73.)

Om bovenstaande inclusie aan te tonen moeten we dus bewijzen dat

$\forall v \in$ U1: {NR} is u.i. in v(GEM).

Stel daarom dat $v \in$ U1, $t \in v$(GEM), $t' \in v$(GEM) en t(NR) = t'(NR); (1)
we moeten dus bewijzen dat $t = t'$, aldus de definitie van unieke identificatie.
Dankzij eis (b) in de definitie van U1 bestaan er tupels p en p' in v(PERS) waarvoor geldt:
t(NR) = p(NR) en t'(NR) = p'(NR) en (2)
t(WPL) = p(WPL) en t'(WPL) = p'(WPL). (3)
Uit (1) en (2) volgt nu dat p(NR) = p'(NR). (4)
Aangezien v(PERS) $\in$ WP, is {NR} u.i. in v(PERS) volgens de definitie van WP. (5)
Uit (4) en (5) volgt nu dat $p = p'$.
Met behulp van (3) kunnen we nu concluderen dat t(WPL) = t'(WPL). (6)
Aangezien v(GEM) $\in$ WG, is {WPL} u.i. in v(GEM).
Uit (6) volgt daarom dat $t = t'$.

Hiermee hebben we (0) bewezen. In feite blijkt dus {NR} een sleutel van GEM in U1 te zijn hoewel {NR} geen sleutel is van de tabellenverzameling WG die tijdens de definitie van U1 is gebruikt! Dergelijke neveneffecten van de andere voorwaarden die gesteld zijn treden overigens in de praktijk wel vaker op (en ook zonder dat men zich daarvan altijd meteen bewust is).

Na dit voorbereidende werk kunnen we nu de berekening van de "kernverzamelingen" Ms(PERS, U1) en Ms(GEM, U1) aan de lezer overlaten; zie Opgave 2.71.

Ter illustratie van Definitie 2.17 besluiten we dit enigszins uitvoerige voorbeeld met de simpele constatering dat bijvoorbeeld {ADR, WPL} $\hat{\rightarrow}$ {PC} in PERS van U1, dit omdat {ADR, WPL} $\hat{\rightarrow}$ {PC} in WP en Un(PERS, U1) = WP.

□ Voorbeeld 2.14.

Bij gegeven DB-skelet blijven permanente afhankelijkheden en sleutels behouden onder "inkrimping" van het DB-universum:

LEMMA 2.19:
Als g een verzamelingsfunctie is en U en U' zijn DB-universa over g en $E \in \mathrm{dom}(g)$ en B en C zijn verzamelingen, dan:
(a) Als $U' \subseteq U$ en $B \hat{\rightarrow} C$ in E van U, dan $B \hat{\rightarrow} C$ in E van U'.
(b) Als $U' \subseteq U$ en B is een sleutel van E in U, dan is B ook een sleutel van E in U'.

Het bewijs van dit eenvoudige lemma laten we in de opgaven aan de lezer over. Ook gaan we daar in op de vraag in hoeverre dit lemma nog kan worden uitgebreid.

OPGAVEN

2.68. In deze opgave gaan we in op de functie v_T over {PERS, GEM} die in Voorbeeld 2.14 is gedefinieerd.

(a) Geef een informele omschrijving van de formele eis (c) in de definitie van v_T.

(b) Bewijs dat voor elke $T \in$ WP de functie v_T inderdaad een element van U1 is.

2.69. Uit de definitie van WG2 in Voorbeeld 2.14 volgt direct dat WG2 $\subseteq$ WG.
In Voorbeeld 2.14 werd verder ook bewezen dat Un(GEM, U1) $\subseteq$ WG2.

(a) Bewijs nu dat WG2 $\subset$ WG.

(b) Bewijs ook dat Un(GEM, U1) $\subset$ WG2.

2.70. Ga na welke van de volgende permanente afhankelijkheden er in Voorbeeld 2.13 gelden en licht uw antwoord telkens kort toe.

(a) {NR} $\hat{\rightarrow}$ {WPL} in PERS van U1;

(b) {NR} $\hat{\rightarrow}$ {WPL} in GEM van U1;

(c) {WPL} $\hat{\rightarrow}$ {NR} in PERS van U1;

(d) {WPL} $\hat{\rightarrow}$ {NR} in GEM van U1.

2.71. In deze opgave gaan we nader in op de minimale sleutels van de in de voorbeelden 2.13 en 2.14 gedefinieerde verzamelingen.

(a) Bepaal Ms(WG) en Ms(WG2).

(b) Bepaal Ms(PERS, U1) en Ms(GEM, U1).

2.72. Bepaal Ms(MEDEW, VBU) en Ms(AFD, VBU), waarbij VBU het in Voorbeeld 1.3 geïntroduceerde DB-universum is.

2.73. In Opgave 2.69 hebben we geconstateerd dat
Un(GEM, U1) $\subset$ WG, ja zelfs dat
Un(GEM, U1) $\subset \{T \in$ WG | {NR} is u.i. in $T\}$.

(a) Probeer Un(GEM, U1) expliciet als deelverzameling van WG te karakteriseren, of althans zo dicht mogelijk "van boven af" te benaderen, dat wil zeggen probeer een zo sterk mogelijke voorwaarde $\phi(T)$ te geven (waarin Un(GEM, U1) zelf niet genoemd wordt) zodanig dat Un(GEM, U1) $\subseteq \{T \in$ WG | $\phi(T)\}$ of zelfs zodanig dat Un(GEM, U1) = $\{T \in$ WG | $\phi(T)\}$.

(b) Bewijs de door u onder (a) gegeven inclusie, c.q. gelijkheid.

2.74. Bewijs Lemma 2.19.

2.75. Blijven bij gegeven DB-skelet behalve de permanente afhankelijkheden en de sleutels ook de minimale sleutels behouden onder inkrimping van het DB-universum?
Geef een bewijs of een tegenvoorbeeld.

2.76. Ga na of permanente afhankelijkheden en sleutels ook behouden blijven onder de andere in Stelling 2.1 genoemde operaties op DB-universa.

2.77. Ga na in hoeverre ook minimale sleutels behouden blijven onder de andere in Stelling 2.1 genoemde operaties op DB-universa.

□

2.3.3 Enige normaalvormen voor database-universa

Een DB-universum U is in Boyce-Codd normaalvorm (respectievelijk derde normaalvorm) als Un(E,U) voor elke tabelindex E van U in BCNF (respectievelijk 3NF) is. Ter onderscheiding van de normaalvormen voor tabellenverzamelingen zullen we voor de normaalvormen voor DB-universa de Nederlandse afkortingen gebruiken:

DEFINITIE 2.20:
Als g een verzamelingsfunctie is en U is een DB-universum over g, dan:

(a) U **is in BCNV** $\stackrel{D}{\Leftrightarrow} \forall E \in \mathrm{He}(U) : \mathrm{Un}(E,U)$ is in BCNF;

(b) U **is in 3NV** $\stackrel{D}{\Leftrightarrow} \forall E \in \mathrm{He}(U) : \mathrm{Un}(E,U)$ is in 3NF.

Lemma 2.20 is een direct gevolg van Lemma 2.18 (omdat we Lemma 2.18 kunnen toepassen op de tabellenverzameling $\mathrm{Un}(E,U)$).

LEMMA 2.20:
Als g een verzamelingsfunctie is en U is een DB-universum over g, dan:
Als U in BCNV is, dan is U ook in 3NV.

OPGAVEN

2.78. Is U1 van Voorbeeld 2.13 in BCNV of in 3NV?

2.79. We definiëren (uitgaande van Voorbeeld 2.13):

$$\mathrm{WP2} = \{T \subseteq \Pi(\mathrm{F4}) \mid \{\mathrm{NR}\} \text{ is u.i. in } T\};$$

$$\begin{aligned} \mathrm{F7} = \{&(\mathrm{PERS} ; \mathrm{WP2}),\\ &(\mathrm{GEM} ; \mathrm{WG})\}; \end{aligned}$$

$$\begin{aligned} \mathrm{U3} = \{v \in \Pi(\mathrm{F7}) \mid\ &\mathrm{id}(\{\mathrm{WPL}\}) \text{ verbindt } v(\mathrm{PERS}) \text{ met } v(\mathrm{GEM}) \text{ en}\\ &\mathrm{id}(\{\mathrm{NR}, \mathrm{WPL}\}) \text{ verbindt } v(\mathrm{GEM}) \text{ met } v(\mathrm{PERS}) \text{ en}\\ &\{(\mathrm{GPL}; \mathrm{WPL})\} \text{ verbindt } v(\mathrm{PERS}) \text{ met } v(\mathrm{GEM})\}. \end{aligned}$$

We wijzen er alvast op dat aan het eind van §2.4 al deze definities nog eens bij elkaar staan in een overzicht van alle definities betreffende onze personen/gemeenten-voorbeelden.

(a) Geef de "nieuwe" eis in de definitie van U3 in woorden weer.

(b) Is U3 in BCNV of in 3NV?

(c) Zij U1 en U3 disjunct? Is de een bevat in de ander?

(d) Bewijs dat $\Pi(\mathrm{F6}) \subseteq \Pi(\mathrm{F7})$.

2.80. Ga na in hoeverre BCNV behouden blijft onder de in Stelling 2.1 genoemde operaties op DB-universa.

2.81. Werk voor 3NV hetzelfde programma af.

□

2.3.4 Permanente verbinding en database-functies

We willen de in §2.1.4 gedefinieerde verbindingsbegrippen ook definiëren op het niveau van DB-universa. We doen dit als volgt:

DEFINITIE 2.21:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $M \in \text{He}(U)$ en $D \in \text{He}(U)$ en h is een functie, dan:

(a) **h verbindt M permanent met D in U** $\overset{D}{\Longleftrightarrow} \forall v \in U : h$ verbindt $v(M)$ met $v(D)$;

(b) **h verbindt M permanent bilateraal met D in U** $\overset{D}{\Longleftrightarrow} \forall v \in U : h$ verbindt $v(M)$ bilateraal met $v(D)$.

VOORBEELD 2.15:

Uit de Voorbeelden 1.3 en 2.13 lezen we de volgende permanente verbindingen in VBU respectievelijk U1 af:

{(AFDNR; ANR)}	verbindt	MEDEW	permanent met	AFD	in VBU;
{(MANNR; NR)}	verbindt	AFD	permanent met	MEDEW	in VBU;
id({NR, WPL})	verbindt	GEM	permanent met	PERS	in U1;
id({WPL})	verbindt	PERS	permanent met	GEM	in U1.

Uit de laatste twee permanente verbindingen kunnen we zelfs een permanente *bilaterale* verbinding afleiden:

id({WPL}) verbindt GEM permanent bilateraal met PERS in U1.

□ Voorbeeld 2.15.

Het volgende begrip leiden we weer in met een voorbeeld.

VOORBEELD 2.16:

Uit de definitie van U1 in Voorbeeld 2.13 kan worden afgeleid dat er voor elke v in U1 een functie H1(v) van v(PERS) naar v(GEM) bestaat die aan elk "persoonstupel" in de DB-toestand v het (eenduidig bepaalde) tupel van diens woonplaats toevoegt. We merken op dat hier in feite sprake is van een (functiewaardige) functie H1 over U1. H1 is een voorbeeld van wat we noemen een *database-functie over* U1 *met betrekking tot* het geordende paar tabelindexen (PERS; GEM).

□ Voorbeeld 2.16.

De algemene definitie van het begrip database-functie luidt als volgt:

DEFINITIE 2.22:

Als g een verzamelingsfunctie is en U is een DB-universum over g en $(M;D) \in \mathrm{He}(U) \times \mathrm{He}(U)$, dan:

H is een **DB-functie over** U **m.b.t.** $(M;D)$ $\overset{D}{\Longleftrightarrow}$ H is een functie over U en

$$\forall v \in U : H(v) \in v(M) \twoheadrightarrow v(D).$$

We noemen M wel *de bronindex van H* en D wel *de doelindex van H*.

Het volgende lemma geeft een in de praktijk belangrijke klasse van DB-functies en berust op Lemma 2.16. Het lemma zegt dat als een functie h de tabelindex M in een DB-universum U permanent verbindt met een tabelindex D, en rng(h) bovendien een sleutel van D in U is, dan is de functie die aan elke $v \in U$ de door h geïnduceerde associatie op $v(M) \times v(D)$ toevoegt een DB-functie over U met betrekking tot het paar $(M;D)$.

LEMMA 2.21:

Als g een verzamelingsfunctie is en U is een DB-universum over g en $M \in \mathrm{He}(U)$ en $D \in \mathrm{He}(U)$ en h is een functie en

(1) rng(h) is een sleutel van D in U en

(2) h verbindt M permanent met D in U,

dan is $\lambda v \in U : \{(t;t') \in v(M) \times v(D) \mid t \restriction \mathrm{dom}(h) = t' \circ h\}$ een DB-functie over U met betrekking tot $(M;D)$.

Bewijs:

We gaan controleren of de bovengenoemde functie

$\lambda v \in U : \{(t;t') \in v(M) \times v(D) \mid t \restriction \mathrm{dom}(h) = t' \circ h\}$ voldoet aan Definitie 2.22.

Tijdens dit bewijs zullen we deze functie H noemen.

Uit de definitie van H is het meteen duidelijk dat H een functie over U is.

Zij nu $v \in U$. Het is duidelijk dat $H(v) \subseteq v(M) \times v(D)$. (a)
Uit (1) volgt dat rng(h) u.i. in $v(D)$ is.
Volgens Lemma 2.16 (3) geldt nu dat $H(v)$ een functie is. (b)
Verder volgt uit (2) dat h de tabel $v(M)$ met de tabel $v(D)$ verbindt.
Volgens Lemma 2.16 (1) geldt nu dat $\mathrm{dom}(H(v)) = v(M)$. (c)
Uit (a), (b) en (c) volgt nu dat $H(v) \in v(M) \twoheadrightarrow v(D)$ voor elke $v \in U$.
Dus H voldoet aan Definitie 2.22.

□

Lemma 2.22 geeft aan hoe we met behulp van *gegeneraliseerde compositie* uit twee gegeven DB-functies een "nieuwe" DB-functie kunnen verkrijgen. Gegeneraliseerde compositie definiëren we voor elk tweetal functiewaardige functies als volgt:

DEFINITIE 2.23:
Als G en H functiewaardige functies zijn, dan:
$G \odot H \stackrel{D}{=} \lambda x \in \mathrm{dom}(G) \cap \mathrm{dom}(H) : G(x) \circ H(x)$.

Met andere woorden $G \odot H$, *de gegeneraliseerde compositie van G na H*, is de functie over $\mathrm{dom}(G) \cap \mathrm{dom}(H)$ gedefinieerd door $(G \odot H)(x) = G(x) \circ H(x)$ voor elke $x \in \mathrm{dom}(G) \cap \mathrm{dom}(H)$. We zullen gegeneraliseerde compositie vooralsnog alleen gebruiken indien $\mathrm{dom}(G) = \mathrm{dom}(H) = U$ waarbij U een of ander DB-universum is. Zo ook in het volgende lemma dat zegt dat de gegeneraliseerde compositie van twee DB-functies met een corresponderende bron- en doelindex weer een DB-functie is:

LEMMA 2.22:
Als U een DB-universum over een verzamelingsfunctie g is en $\{M,D,D'\} \subseteq \mathrm{He}(U)$ en
(1) H is een DB-functie over U m.b.t. $(M; D)$ en
(2) G is een DB-functie over U m.b.t. $(D; D')$,
dan is $G \odot H$ een DB-functie over U m.b.t. $(M; D')$.

OPGAVEN

2.82. Verbindt $\{(\mathrm{GPL};\ \mathrm{WPL})\}$ de tabelindex PERS permanent bilateraal met GEM in U3 van Opgave 2.79? Geef een bewijs of een tegenvoorbeeld.

2.83. (a) Geef een formele definitie van de in Voorbeeld 2.16 informeel omschreven functie H1.

(b) Is H1 van de in Lemma 2.21 beschreven vorm?

2.84. We definiëren de functies H2 en H3 als volgt:

$$\mathrm{H2} = \lambda v \in \mathrm{U1} : \{(t\,; t') \in v(\mathrm{GEM}) \times v(\mathrm{PERS}) \mid t(\mathrm{NR}) = t'(\mathrm{NR})\} \text{ en}$$
$$\mathrm{H3} = \lambda v \in \mathrm{U1} : \{(t\,; t') \in v(\mathrm{PERS}) \times v(\mathrm{GEM}) \mid t(\mathrm{GPL}) = t'(\mathrm{WPL})\},$$

waarbij U1 het in Voorbeeld 2.13 gedefinieerde DB-universum is.
Geef een informele beschrijving van H2 en H3. Zijn H2 en H3 DB-functies?
Zo nee, waarom niet?

2.85. Bewijs het volgende:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $(M\,;D) \in \mathrm{He}(U) \times \mathrm{He}(U)$ en $a \in g(M)$ en $a' \in g(D)$ en $\{a'\}$ is een sleutel van D in U en $\{t(a) \mid t \in v(M)\} \subseteq \{t'(a') \mid t' \in v(D)\}$ voor elke $v \in U$,
dan is $\lambda v \in U : \{(t\,;t') \in v(M) \times v(D) \mid t(a) = t'(a')\}$ een DB-functie over U met betrekking tot $(M\,;D)$.

2.86. Als g een verzamelingsfunctie is en U een DB-universum over g en $M \in \mathrm{He}(U)$ en $D \in \mathrm{He}(U)$ en $g(M) \cap g(D)$ is een sleutel van D in U en $\mathrm{id}(g(M) \cap g(D))$ verbindt $v(M)$ met $v(D)$ voor elke $v \in U$, en H is gedefinieerd door

$$H = \lambda v \in U : \{(t\,;t') \in v(M) \times v(D) \mid t \text{ en } t' \text{ stemmen overeen op } g(M) \cap g(D)\},$$

dan:
(a) H is een DB-functie over U m.b.t. $(M\,;D)$ en
(b) $v(M) \bowtie v(D) = \{t \cup H(v)\,(t) \mid t \in v(M)\}$ voor elke $v \in U$.
Bewijs dit.

2.87. Bewijs Lemma 2.22.

2.88. Is gegeneraliseerde compositie associatief?

2.89. Zijn H2 en H3 uit Opgave 2.84 functiewaardige functies?
Zo ja, geef dan een informele beschrijving van:

$$\mathrm{H2} \odot \mathrm{H1},\ \mathrm{H1} \odot \mathrm{H2},\ \mathrm{H2} \odot \mathrm{H3},\ \mathrm{H3} \odot \mathrm{H2} \text{ en } \mathrm{H3} \odot (\mathrm{H2} \odot \mathrm{H1}),$$

waarbij H1 de in Voorbeeld 2.16 beschreven functie is.
Welke hiervan zijn DB-functies over U1, en met betrekking tot welke indexparen?

2.90. We definiëren (uitgaande van Voorbeeld 2.13):

$$
\begin{aligned}
\text{WPE} &= \{T \in \text{WP} \mid T \text{ is eindig}\};\\
\text{WGE} &= \{T \in \text{WG} \mid T \text{ is eindig}\};\\
\text{F8} &= \{(\text{PERS} \ ; \text{WPE}),\\
&\qquad (\text{GEM} \ ; \text{WGE})\}.
\end{aligned}
$$

Deze definities staan ook in het overzicht aan het eind van §2.4, samen met de definities uit Voorbeeld 2.13.

(a) Bewijs dat $\Pi(\text{F8}) \subseteq \Pi(\text{F6})$.

(b) Bewijs dat WGE = WG.

(c) Is WGE een eindige verzameling?

(d) Bewijs dat WPE $\subset$ WP.

2.91. Onder verwijzing naar de Opgaven 2.79, 2.84 en 2.90 (of het overzicht aan het eind van §2.4) definiëren we verder nog:

$$
\begin{aligned}
&\text{U4} = \text{U3} \cap \Pi(\text{F8}),\\
&\text{H4} = \text{H2} \upharpoonright \text{U4 en}\\
&\text{H5} = \text{H3} \upharpoonright \text{U4}.
\end{aligned}
$$

(a) Bewijs dat U4 $\subseteq$ U1.

(b) Geef een informele beschrijving van H4 en H5.

(c) Zijn H4 en H5 ook DB-functies?

2.92. Uitgaande van Opgave 2.91 definiëren we (op recursieve wijze):

$$G_0 = \text{H4} \odot \text{H5} \text{ en } G_{n+1} = G_n \odot G_0 \text{ voor elke } n \in \mathbb{N}.$$

(a) Geef een informele beschrijving van G_0.

(b) Probeer voor willekeurige $n \in \mathbb{N}$ een informele beschrijving van G_n te geven.

(c) Bewijs dat voor elke $n \in \mathbb{N}$ de functie G_n een DB-functie over U4 is.

(d) Bewijs dat voor elke $v \in$ U4 niet alle $G_n(v)$ verschillend zijn (m.a.w., $\forall v \in \mathrm{U4} : \exists n \in \mathbb{N} : \exists m \in \mathbb{N} : n \neq m$ en $G_n(v) = G_m(v)$).

(e) Ga na of alle G_n verschillend zijn.

□

2.4 SAMENVATTING

In Figuur 2.10 hebben we de in de hoofdstukken 1 en 2 ingevoerde begrippen bij elkaar gezet, gegroepeerd rondom de vier hoofdbegrippen *tabel, tabellenverzameling, DB-toestand* en *DB-universum.* De begrippen zijn op twee "orthogonale" manieren in tweeën verdeeld:

(1) Enerzijds is er een scheidslijn tussen

   (a) die begrippen die in feite reeds behoren tot het gebied *bestandstheorie* (dat wil zeggen de theorie van afzonderlijke bestanden), en

   (b) die begrippen die pas een rol spelen bij *database-theorie* (dat wil zeggen de theorie van verschillende "geïntegreerde" bestanden).

(2) Anderzijds is er een scheidslijn tussen

   (a) die begrippen die betrekking hebben op een *toestand*, en

   (b) die begrippen die betrekking hebben op een *toestandsruimte.*

Dus, uitgaande van een *tabel* over een verzameling A kunnen we in twee "dimensies" uitbreiden:

I: enerzijds naar verschillende soorten tabellen "tegelijkertijd"
(dat wil zeggen naar een *DB-toestand* over een verzamelingsfunctie g),
wat we bij de overgang van §1.1 naar §1.2 hebben gedaan, en

II: anderzijds naar verschillende tabellen van dezelfde "soort"
(dat willen zeggen naar een *tabellenverzameling* over A),
wat we bij de overgang van §2.1 naar §2.2 hebben gedaan.

Wanneer we in *beide* "dimensies" uitbreiden, dan komen we tot een *DB-universum* over g. In Hoofdstuk 1 hebben we dat bereikt door eerst uitbreiding I uit te voeren (van §1.1 naar §1.2) en daarna uitbreiding II (van §1.2 naar §1.3). Bij de opbouw van Hoofdstuk 2 daarentegen hebben we eerst uitbreiding II uitgevoerd (van §2.1 naar §2.2) en daarna uitbreiding I (van §2.2 naar §2.3).

	Bestand	Database
Toestand (momentaan) (incidenteel) (extensioneel)	§1.1: T is een *tabel* over A §2.1.1: $T \upharpoonright B$, $T \bowtie T'$, $T \infty h$ §2.1.2: $B \to C$ in T §2.1.3: B is (minimaal) u.i. in T §2.1.4: h verbindt T (bilateraal) met T'	§1.2: v is een *DB-toestand* over g
Toestandsruimte (permanent) (structureel) (intensioneel)	§2.2: W is een *tabellenverzameling* over A §2.2.1: $B \hat{\to} C$ in W §2.2.2: B is een (minimale) sleutel van W Head(W), Ms(W) Prim(W), Sec(W) §2.2.3: W is in BCNF W is in 3NF	§1.3: U is een *DB-universum* over g §2.3.2: Un(E,U) $B \hat{\to} C$ in E van U B is een (minimale) sleutel van E in U Head(E,U), Ms(E,U) Prim(E,U), Sec(E,U) §2.3.3: U is in BCNV U is in 3NV §2.3.4: h verbindt M permanent (bilateraal) met D in U H is een DB-functie over U m.b.t. (M ; D) $G \odot H$

Figuur 2.10: Overzicht van de ingevoerde begrippen

Als een soort referentiekader geven we als laatste voorbeeld in dit hoofdstuk nog het simpele maar in de database-literatuur zeer bekende en veel gebruikte "suppliers/parts/shipments"-voorbeeld (zie onder andere [Da 86]).

VOORBEELD 2.17:

Het suppliers/parts/shipments-voorbeeld is gebaseerd op het volgende DB-skelet, dat we g2 zullen noemen:

g2 = {(S ; {S#, SNAME, STATUS, CITY}),	suppliers
(P ; {P#, PNAME, COLOR, WEIGHT, CITY}),	parts
(SP; {S#, P#, QTY})}	shipments

Hierbij staat S# voor *supplier number*, P# voor *part number* en QTY voor *quantity*.

We definiëren achtereenvolgens drie objectkarakteriseringen, één database-karakterisering en ten slotte het DB-universum.

FS = {(S# ; Chs(5)),
(SNAME ; Chs(20)),
(STATUS; Int(4)),
(CITY ; Chs(15))};

FP = {(P# ; Chs(6)),
(PNAME ; Chs(20)),
(COLOR ; Chs(6)),
(WEIGHT ; Int(4)),
(CITY ; Chs(15))};

FSP = {(S# ; Chs(5)),
(P# ; Chs(6)),
(QTY; Int(9))};

SPSK = {(S ; $\{T \subseteq \Pi(\mathrm{FS}) \mid \{\mathrm{S\#}\}$ is u.i. in $T\}$),
(P ; $\{T \subseteq \Pi(\mathrm{FP}) \mid \{\mathrm{P\#}\}$ is u.i. in $T\}$),
(SP; $\{T \subseteq \Pi(\mathrm{FSP}) \mid \{\mathrm{S\#}, \mathrm{P\#}\}$ is u.i. in $T\}$)};

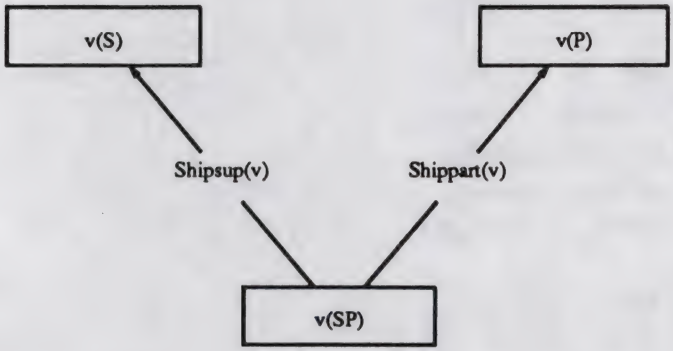
$$\begin{aligned}\text{SPSU} = \{v \mid\ & v \in \Pi(\text{SPSK}) \text{ en} \\ & \{(\text{S\#}\,;\,\text{S\#})\} \text{ verbindt } v(\text{SP}) \text{ met } v(\text{S}) \text{ en} \\ & \{(\text{P\#}\,;\,\text{P\#})\} \text{ verbindt } v(\text{SP}) \text{ met } v(\text{P})\}.\end{aligned}$$

We kunnen nu de volgende functies over SPSU definiëren:

$$\text{Shipsup} = \lambda v \in \text{SPSU} : \{(t\,;\,t') \in v(\text{SP}) \times v(\text{S}) \mid t(\text{S\#}) = t'(\text{S\#})\}$$
$$\text{Shippart} = \lambda v \in \text{SPSU} : \{(t\,;\,t') \in v(\text{SP}) \times v(\text{P}) \mid t(\text{P\#}) = t'(\text{P\#})\}$$

Volgens Lemma 2.21 is Shipsup een DB-functie over SPSU met betrekking tot het tabelindexpaar (SP; S) en Shippart een DB-functie over SPSU met betrekking tot het paar (SP; P). Voor $v \in$ SPSU is Shipsup(v) $\in v(\text{SP}) \twoheadrightarrow v(\text{S})$ de functie die aan elk shipment-tupel in toestand v het "bijbehorende" supplier-tupel toevoegt en is Shippart(v) $\in v(\text{SP}) \twoheadrightarrow v(\text{P})$ de functie die aan elk shipment-tupel in toestand v het "bijbehorende" part-tupel toevoegt.

We kunnen de zojuist genoemde constateringen dat Shipsup(v) $\in v(\text{SP}) \twoheadrightarrow v(\text{S})$ en Shippart(v) $\in v(\text{SP}) \twoheadrightarrow v(\text{P})$ als volgt schematisch weergeven:



Figuur 2.11: Enige functionele verbanden

Vooral wanneer we te maken hebben met een gróót aantal DB-functies - wat in de praktijk vaak het geval is - dan kan zo'n informeel plaatje nog wel eens handig zijn. Wellicht ten overvloede merken we op dat omgekeerd zo'n informele schets in het algemeen nog niet voldoende is om daaruit de precieze definities van de aangegeven functies te kunnen afleiden.

□ Voorbeeld 2.17.

Gemakshalve geven we ook nog een overzicht van onze (cumulatieve) personen/gemeenten-voorbeelden, die zijn verspreid over de voorbeelden 2.8, 2.13 en 2.14 en over op opgaven 2.79, 2.84, 2.90 en 2.91.

$$\begin{aligned}
\mathrm{F1} &= \{(\mathrm{STR}\ ;\ \mathrm{Chs}(50)),\\
&\quad\ (\mathrm{HNR}\ ;\ [1\ ..\ 5000])\};\\
\mathrm{F2} &= \{(\mathrm{NR}\ ;\ [1000\ ..\ 9999]),\\
&\quad\ (\mathrm{LC}\ ;\ \mathrm{Chs}(2))\};\\
\mathrm{F3} &= \{(\mathrm{DG}\ ;\ [1\ ..\ 31]),\\
&\quad\ (\mathrm{MND}\ ;\ [1\ ..\ 12]),\\
&\quad\ (\mathrm{JR}\ ;\ \mathbb{N})\};
\end{aligned}$$

$$\begin{aligned}
\mathrm{F4} &= \{(\mathrm{NR}\ ;\ \mathbb{N}-\{0\}),\\
&\quad\ (\mathrm{NM}\ ;\ \mathrm{Chs}(40)),\\
&\quad\ (\mathrm{ADR}\ ;\ \Pi(\mathrm{F1})),\\
&\quad\ (\mathrm{PC}\ ;\ \Pi(\mathrm{F2})),\\
&\quad\ (\mathrm{WPL}\ ;\ \mathrm{Chs}(30)),\\
&\quad\ (\mathrm{GBD}\ ;\ \Pi(\mathrm{F3})),\\
&\quad\ (\mathrm{GPL}\ ;\ \mathrm{Chs}(30)),\\
&\quad\ (\mathrm{NMK};\ \mathbb{P}(\mathrm{Chs}(30)))\};
\end{aligned}$$

$$\begin{aligned}
\mathrm{S2} &= \{\text{'GR'}, \text{'FR'}, \text{'DR'},\\
&\quad\ \text{'OV'}, \text{'FL'}, \text{'GE'},\\
&\quad\ \text{'UT'}, \text{'NH'}, \text{'ZH'},\\
&\quad\ \text{'ZE'}, \text{'NB'}, \text{'LI'}\};\\
\mathrm{F5} &= \{(\mathrm{WPL}\ ;\ \mathrm{Chs}(30)),\\
&\quad\ (\mathrm{PROV}\ ;\ \mathrm{S2}),\\
&\quad\ (\mathrm{INWA}\ ;\ \mathbb{N}),\\
&\quad\ (\mathrm{OPVL}\ ;\ \mathbb{N}),\\
&\quad\ (\mathrm{NR}\ ;\ \mathbb{N})\};
\end{aligned}$$

$$\begin{aligned}
\mathrm{WP} &= \{T \subseteq \Pi(\mathrm{F4}) \mid \{\mathrm{NR}\} \text{ is u.i. in } T \text{ en}\\
&\qquad \{\mathrm{PC}\} \to \{\mathrm{WPL}\} \text{ in } T \text{ en}\\
&\qquad \{\mathrm{ADR}, \mathrm{WPL}\} \to \{\mathrm{PC}\} \text{ in } T \text{ en}\\
&\qquad \forall t \in T\text{: als } t(\mathrm{NR}) \neq 1 \text{ dan } \exists t' \in T : t'(\mathrm{NR}) = t(\mathrm{NR}) - 1\};\\
\mathrm{WG} &= \{T \subseteq \Pi(\mathrm{F5}) \mid \{\mathrm{WPL}\} \text{ is u.i. in } T\};\\
\mathrm{WP2} &= \{T \subseteq \Pi(\mathrm{F4}) \mid \{\mathrm{NR}\} \text{ is u.i. in } T\};\\
\mathrm{WG2} &= \{T \in \mathrm{WG} \mid \{\mathrm{NR}\} \text{ is u.i. in } T\};\\
\mathrm{WPE} &= \{T \in \mathrm{WP} \mid T \text{ is eindig}\};\\
\mathrm{WGE} &= \{T \in \mathrm{WG} \mid T \text{ is eindig}\};
\end{aligned}$$

$$\begin{aligned}
\mathrm{F6} &= \{(\mathrm{PERS};\ \mathrm{WP}),\ (\mathrm{GEM};\ \mathrm{WG})\};\\
\mathrm{F7} &= \{(\mathrm{PERS};\ \mathrm{WP2}),\ (\mathrm{GEM};\ \mathrm{WG})\};\\
\mathrm{F8} &= \{(\mathrm{PERS};\ \mathrm{WPE}),\ (\mathrm{GEM};\ \mathrm{WGE})\};
\end{aligned}$$

$$\begin{aligned}
\mathrm{U1} &= \{v \in \Pi(\mathrm{F6}) \mid \mathrm{id}(\{\mathrm{WPL}\}) \text{ verbindt } v(\mathrm{PERS}) \text{ met } v(\mathrm{GEM}) \text{ en}\\
&\qquad \mathrm{id}(\{\mathrm{NR}, \mathrm{WPL}\}) \text{ verbindt } v(\mathrm{GEM}) \text{ met } v(\mathrm{PERS})\};\\
\mathrm{U3} &= \{v \in \Pi(\mathrm{F7}) \mid \mathrm{id}(\{\mathrm{WPL}\}) \text{ verbindt } v(\mathrm{PERS} \text{ met } v(\mathrm{GEM}) \text{ en}\\
&\qquad \mathrm{id}(\{\mathrm{NR}, \mathrm{WPL}\}) \text{ verbindt } v(\mathrm{GEM}) \text{ met } v(\mathrm{PERS}) \text{ en}\\
&\qquad \{(\mathrm{GPL};\ \mathrm{WPL})\} \text{ verbindt } v(\mathrm{PERS}) \text{ met } v(\mathrm{GEM})\};\\
\mathrm{U4} &= \mathrm{U3} \cap \Pi(\mathrm{F8});
\end{aligned}$$

$$\begin{aligned}
\mathrm{H2} &= \lambda v \in \mathrm{U1}\text{: } \{(t\ ; t') \in v(\mathrm{GEM}) \times v(\mathrm{PERS}) \mid t(\mathrm{NR}) = t'(\mathrm{NR})\};\\
\mathrm{H3} &= \lambda v \in \mathrm{U1}\text{: } \{(t\ ; t') \in v(\mathrm{PERS}) \times v(\mathrm{GEM}) \mid t(\mathrm{GPL}) = t'(\mathrm{WPL})\};\\
\mathrm{H4} &= \mathrm{H2} \upharpoonright \mathrm{U4};\\
\mathrm{H5} &= \mathrm{H3} \upharpoonright \mathrm{U4}.
\end{aligned}$$

We besluiten Hoofdstuk 2 met enige herhalingsopgaven.

OPGAVEN

2.93. We definiëren de zogeheten *(left) outer join* van een tabel T over een verzameling A met een tabel T' over een verzameling A' als volgt:

$$T \mathbin{\triangleright\!\!<} T' \overset{D}{=} (T \bowtie T') \cup (T - ((T \bowtie T') \upharpoonright\!\!\upharpoonright A)).$$

(a) Is de verzameling $T \mathbin{\triangleright\!\!<} T'$ altijd, dat wil zeggen voor alle tabellen T en T', formeel weer een tabel? Licht uw antwoord toe met een bewijs of een tegenvoorbeeld.

(b) Als $T \upharpoonright\!\!\upharpoonright A' \subseteq T' \upharpoonright\!\!\upharpoonright A$ dan $T \mathbin{\triangleright\!\!<} T' = T \bowtie T'$. Bewijs dit.

(c) Is de left outer join commutatief?
(Met andere woorden, geldt altijd $T \mathbin{\triangleright\!\!<} T' = T' \mathbin{\triangleright\!\!<} T$?)

In Figuur 2.2 in Voorbeeld 2.2 zijn de tabellen V1 en V2 weergegeven.

(d) Geef het aantal elementen van de verzamelingen V1 $\mathbin{\triangleright\!\!<}$ V2 en V2 $\mathbin{\triangleright\!\!<}$ V1.

(e) Probeer ook V1 $\mathbin{\triangleright\!\!<}$ V2 en V2 $\mathbin{\triangleright\!\!<}$ V1 in een figuur weer te geven.

2.94. We definiëren:

(D1) Als A en B verzamelingen zijn, $c \notin (A - B)$ en T is een tabel over A, dan:
$\text{Pack}_{B,c}(T) \overset{D}{=} \{t \not\upharpoonright B \cup \{(c\,; t \upharpoonright B)\} \mid t \in T\}$

(D2) Als A en B verzamelingen zijn, $a \in A$, $B \cap (A - \{a\}) = \varnothing$, T is een tabel over A en $\forall t \in T : t(a)$ is een functie over B, dan:
$\text{Unpack}_a(T) \overset{D}{=} \{t \not\upharpoonright \{a\} \cup t(a) \mid t \in T\}$.

Beantwoord de volgende vragen.

(a) Zijn $\text{Pack}_{B,c}(T)$ en $\text{Unpack}_a(T)$ weer tabellen?
Zo ja, over welke verzamelingen?

Bewijs de volgende twee gelijkheden en geef in elk van de beide gevallen de precieze voorwaarden (voor a, B, T, etcetera) aan.

(b) $\text{Unpack}_a(\text{Pack}_{B,a}(T)) = T$.

(c) $\text{Pack}_{B,a}(\text{Unpack}_a(T)) = T$.

2.95. Geldt er gelijkheid in Opgave 2.17(b)? Geef een bewijs of een tegenvoorbeeld.

2.96. Laat A, B en C verzamelingen zijn en T een tabel over A. We definiëren:
$B \twoheadrightarrow C$ in $T \overset{D}{\Longleftrightarrow} \forall t \in T : \forall t' \in T$: als $t \restriction B = t' \restriction B$ dan $t\,\theta(t' \restriction C) \in T$.

(We noemen dan C *meerwaardig afhankelijk* van B in T;
in het Engels: C is *multivalued dependent* on B within T.)

(a) Als $B \subseteq A$ en $C \subseteq A$ en $B \to C$ in T dan $B \twoheadrightarrow C$ in T.
Bewijs dit.

(b) Geef een voorbeeld van een meerwaardige afhankelijkheid die niet een momentane afhankelijkheid is.

(c) Bewijs de volgende equivalentie:
$B \twoheadrightarrow C$ in $T \Longleftrightarrow T = T \restriction (B \cup C) \bowtie T \restriction (B \cup (A - C))$.

2.97. Als A, A' en A'' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en T'' is een tabel over A'' en h en h' zijn injectieve functies met $\mathrm{dom}(h) \subseteq A$ en $\mathrm{rng}(h) \subseteq \mathrm{rng}(h')$ en $\mathrm{dom}(h') \subseteq A'$ en $\mathrm{rng}(h') \subseteq A''$ en h verbindt T met T' en h' verbindt T' met T'', dan verbindt $h' \circ h$ de tabel T met T''.
Bewijs dit of geef een tegenvoorbeeld.

2.98. Als A en A' verzamelingen zijn en T is een tabel over A en T' is een tabel over A' en h is een injectieve functie en $\mathrm{dom}(h) \subseteq A$ en $\mathrm{rng}(h) \subseteq A'$ dan:

h verbindt T met $T' \Longleftrightarrow T = T \bowtie (T' \infty h)$.

Bewijs dit.

2.99. Als T en T' verzamelingen functies zijn en C is een verzameling en h is een injectieve functie, dan:

(a) $\mathrm{id}(C)$ verbindt T bilateraal met $T' \Longleftrightarrow \mathrm{id}(C)$ verbindt T' bilateraal met T;

(b) h verbindt T bilateraal met $T' \Longleftrightarrow h^{-1}$ verbindt T' bilateraal met T.

Bewijs dit.

2.100. In de (SQL-)praktijk moet men voor gegeven attributentransformatie h en tabellen T en T' vaak de volgende equivalentie gebruiken:

$$h \text{ verbindt } T \text{ met } T' \iff 0 = |\{t \in T \mid 0 = |\{t' \in T' \mid t \restriction \text{dom}(h) = t' \circ h\}|\}|.$$

Bewijs dit gezochte gedrocht.

2.101. Bepaal de verzameling van alle minimale sleutels van de verzameling W3 bestaande uit de volgende 3 tabellen:

T21:

$a1$	$a2$	$a3$	$a4$	$a5$
1	4	9	p	s
1	4	9	q	t

T22:

$a1$	$a2$	$a3$	$a4$	$a5$
1	4	7	p	s
1	6	7	p	s
1	4	9	p	s
2	5	9	q	t
3	5	9	p	t
3	4	7	p	s

T23:

$a1$	$a2$	$a3$	$a4$	$a5$
2	6	7	p	t
2	5	7	p	s
1	4	7	q	t
1	4	9	q	t
2	5	9	p	s

Licht uw antwoord kort toe.

2.102. Als A en B verzamelingen zijn en W is een verzameling tabellen over A, geldt dan

B is een minimale sleutel van $W \iff \forall T \in W : B$ is minimaal u.i. in T?

Onderzoek beide implicaties afzonderlijk op hun correctheid.

2.103. We definiëren:

Als A een verzameling is en W is een verzameling tabellen over A, dan:

W **is in 2NF** $\overset{D}{\iff} \forall a \in \text{Sec}(W) : \forall S \in \text{Ms}(W) : \forall B \subseteq S$: als $B \hat{\rightarrow} \{a\}$ in W dan $B = S$.

Dus, informeel verwoord, W is in 2NF (*tweede normaalvorm*) desda elke secundair attribuut "volledig afhankelijk" is van elke minimale sleutel van W.

(a) Als W in 3NF is dan is W in 2NF. Bewijs dit.

(b) Is WP uit Voorbeeld 2.8 in 2NF?

2.104. Deze opgave is een vervolg op Opgave 2.96. We introduceren hier het begrip meerwaardige afhankelijkheid (multivalued dependency) op het niveau van tabellenverzamelingen. Ook voeren we in deze opgave de *vierde normaalvorm* in.

Als A, B en C verzamelingen zijn en W is een verzameling tabellen over A, dan definiëren we:

(D1) $B \;\hat{\twoheadrightarrow}\; C$ **in** $W \;\overset{D}{\Longleftrightarrow}\; \forall T \in W : B \twoheadrightarrow C$ in T.

(D2) W **is in 4NF** $\;\overset{D}{\Longleftrightarrow}\; \forall X \subseteq \text{Head}(W) : \forall Y \subseteq \text{Head}(W)$:
als $X \;\hat{\twoheadrightarrow}\; Y$ in W en $Y \not\subseteq X$ en $X \cup Y \subset \text{Head}(W)$
dan is X een sleutel van W.

(a) Als W in 4NF is, dan is W ook in BCNF. Bewijs dit.

(b) Is $\varnothing$ in 4NF?

(c) Is WA uit Voorbeeld 1.3 in 4NF?

2.105. Definieer naar aanleiding van het DB-universum in Voorbeeld 2.13 een DB-universum UNED dat wél precies de Nederlandse situatie weerspiegelt.

□

3 CONSTRUCTIE VAN DATABASE-UNIVERSA

In Hoofdstuk 1 hebben we gedefinieerd *wat* een DB-universum in theorie is. In dit hoofdstuk zullen we aangeven *hoe* een DB-universum in de praktijk kan worden geconstrueerd.

Definities van DB-universa plegen in de praktijk nogal omvangrijk te zijn. Het is bijvoorbeeld niet ongebruikelijk dat er ten behoeve van de werkzaamheden van één enkele afdeling binnen een organisatie wel een paar honderd attributen nodig zijn, dit dus nog afgezien van de beschrijving van de vele *eisen* waaraan de gegevens dienen te voldoen. Het is daarom van belang om constructies van DB-universa op een stelselmatige wijze op te bouwen. Bovendien is het verstandig om dit bij voorkeur op een *modulaire* wijze te doen aangezien (gedeelten van) bestaande DB-universa nog wel eens aan verandering onderhevig willen zijn.

Wij zullen in de constructie van een DB-universum over een *gegeven* DB-skelet g vier "fasen" onderscheiden en aan elke fase een paragraaf wijden. (Het bepalen van het DB-skelet zelf vormt nog een fase apart; zie §3.0.) Deze vier fasen bestaan uit het bepalen van

(1) de verzameling toegestane **attribuutwaarden** *per* attribuut *per* tabelindex,

(2) de verzameling toegestane **tupelwaarden** *per* tabelindex,

(3) de verzameling toegestane **tabelwaarden** *per* tabelindex, en ten slotte

(4) de verzameling toegestane **database-waarden**.

VOORBEELD 3.1:

In Voorbeeld 1.3 zijn de betreffende verzamelingen (uitgaande van het eerder in Voorbeeld 1.2 reeds gegeven DB-skelet g1):

(1) de verzameling FM(a) voor elk attribuut a horend bij de tabelindex MEDEW en de verzameling FA(a) voor elk attribuut a horend bij de tabelindex AFD,

(2) Π(FM) voor MEDEW en Π(FA) voor AFD,

(3) WM voor MEDEW en WA voor AFD, en ten slotte

(4) VBU.

□ Voorbeeld 3.1.

De waardenverzamelingen worden in feite bepaald door de eisen waaraan de gegevens in alle mogelijke toestanden volgens de organisatie in kwestie dienen te voldoen. Deze eisen aan de mogelijke *toestanden* worden in de literatuur wel *statische constraints* genoemd. Behalve statische constraints zullen we in Hoofdstuk 5 ook *dynamische constraints* behandelen, dat zijn de eisen waaraan toestands*overgangen* dienen te voldoen.

Dergelijke eisen, die in dit verband ook wel "business rules" worden genoemd, weerspiegelen in wezen voor een belangrijk gedeelte de "semantiek" van de gegevens voor de organisatie in kwestie! Eén van de doelstellingen van dit boek is dan ook de lezer te laten zien hoe deze semantiek formeel kan worden gespecificeerd.

Zoals reeds beschreven in bijvoorbeeld [Re 82] en [Br 80] kunnen we parallel aan de eerder genoemde fasering van de constructie van een DB-universum ook vier soorten statische constraints onderscheiden:

(1) **Attribuutconstraints**; dit zijn voorwaarden waaraan de waarden van de afzonderlijke attributen moeten voldoen.

(2) **Tupelcontraints** (of "inter-attribuut" constraints); dit zijn voorwaarden waaraan combinaties van attribuutwaarden binnen hetzelfde tupel moeten voldoen.

(3) **Tabelconstraints** (of "inter-tupel" constraints); dit zijn voorwaarden waaraan combinaties van tupels binnen dezelfde tabel moeten voldoen.

(4) **Databaseconstraints** (of "inter-tabel" constraints); dit zijn voorwaarden waaraan combinaties van tabellen binnen dezelfde DB-toestand moeten voldoen.

Deze vier soorten zijn dus geordend naar toenemende "reikwijdte".

We zullen in elk van de te bespreken fasen de behandelde stof larderen met verwijzingen naar de in de vorige hoofdstukken gegeven voorbeelden en ook alvast naar het grote voorbeeld in Hoofdstuk 4. Daarnaast zullen we de behandelde stof ook telkens nog toelichten aan de hand van een klein, maar gevarieerd modelleringsvoorbeeld. Dit "Madurodam van de gegevensmodellering" wordt daarbij stap voor stap opgebouwd.

3.0 HET DATABASE-SKELET

In feite is er ook nog een "nulde" fase, namelijk de fase waarin de relevante tabelindexen en de relevante attributen per tabelindex - kortom het juiste DB-skelet *g* - worden bepaald. In de praktijk wordt deze fase van het ontwerpproces nogal eens onderschat, zowel qua moeilijkheidsgraad als qua tijdsduur. De belangrijkste reden hiervoor is dat de toekomstige gebruikers van de database bij nader inzien vaak zelf ook niet precies weten wat de feitelijke structuur van hun organisatie en hun gegevens is, en wat daarvan wel en wat niet relevant is.

Tot nu toe hebben we één expliciet voorbeeld van een DB-skelet genoemd, en wel het DB-skelet g1 in Voorbeeld 1.2:

g1 = {(AFD ; {ANR, MANNR, NAAM}),
(MEDEW ; {AFDNR, NR, NAAM, GESL, SAL})}

Het DB-skelet g1 kent zowel zogeheten *homonieme* attributen als zogeheten *synonieme* attributen:

- De attributen NAAM van AFD en NAAM van MEDEW zijn "homoniem" (dat wil zeggen gelijknamig maar met verschillende "betekenissen").
- De attributen ANR van AFD en AFDNR van MEDEW zijn "synoniem" (dat wil zeggen ongelijknamig maar met dezelfde "betekenis").

De volgende verzamelingsfunctie, g2, is het gemeenschappelijke DB-skelet van de DB-universa U1, U3 en U4 zoals die in het overzicht aan het eind van §2.4 voorkomen:

g2 = {(PERS ; {NR, NM, ADR, PC, WPL, GBD, GPL, NMK}),
(GEM ; {WPL, PROV, INWA, OPVL, NR})}.

Het DB-skelet van het DB-universum UZKH in §4.2 bestaat uit 19 tabelindexen met in totaal 128 attributen. Voorbeelden van tabelindexen zijn hier PAT, MW, MED en MVS.

VOORBEELD 3.2:

In een produktiebedrijf wil men gegevens bijhouden over de produkten die men zelf maakt en over de basisprodukten die men daarvoor inkoopt. Bovendien wil men zogeheten *stuklijsten* bijhouden, dat wil zeggen lijsten die de directe onderdelen van samengestelde produkten opsommen en daarbij tevens aangeven hoe vaak dat onderdeel als direct onderdeel in dat produkt optreedt. Een stuklijst wordt ook wel een Bill Of Material (of kortweg BOM) genoemd. Stuklijsten spelen een centrale rol in

produktiebedrijven en geven bovendien aanleiding tot allerlei interessante wiskundige vragen. Deze vragen zijn vaàk recursief van aard.

Van elk *produkt* wil men in het onderhavige produktiebedrijf het *produktnummer*, de *omschrijving*, de *produktsoort*, de *kostprijs* per stuk, de *verkoopprijs* per stuk, de *winst* per stuk en de aanwezige *voorraad* bijhouden. (Alle prijzen zijn in centen uitgedrukt.) Bovendien wil men van elk produkt weten of het een *samengesteld* produkt dan wel een basisprodukt is.

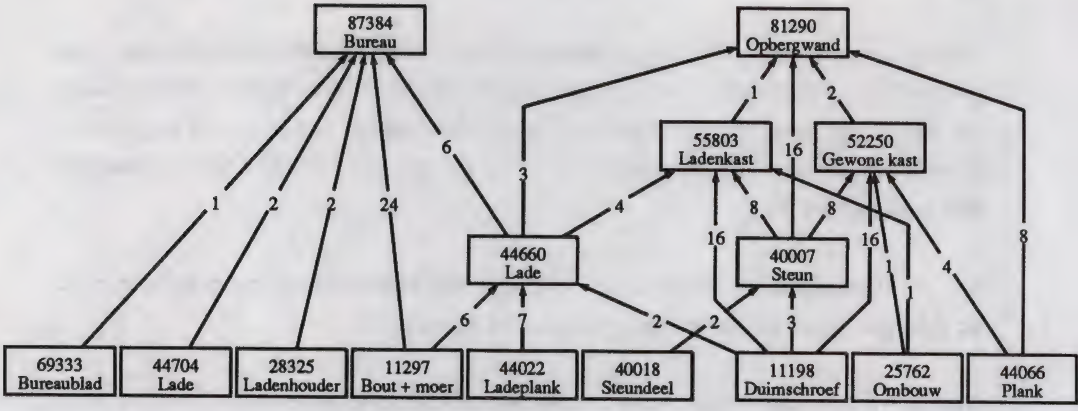
Van elk *basisprodukt* wil men verder bovendien nog bijhouden wat het *gewicht* is en of de gebruikte *gewichtseenheid* nu gram dan wel kilogram is.

Verder wil men dus weten welke *produkten* welke directe *onderdelen* bevatten en in welk *aantal stuks*. Dit gebeurt op basis van zogeheten *stuklijstregels*.

Dit brengt ons tot het volgende DB-skelet:

GBP = {(PROD ; {PNR, OMS, SRT, KPR, VKP, WNST, VRD, SAM}),
(BAS ; {PNR, GW, GWE}),
(BOM ; {PNR, ONR, AST})}.

Ter illustratie geeft Figuur 3.2 een representatieve DB-toestand v2 over GBP weer. We geven in Figuur 3.1 echter eerst een grafische weergave van de (assemblage)gegevens die in v2(BOM) voorkomen, aangevuld met de omschrijvingen uit v2(PROD). Het laat daarmee de hiërarchische produktopbouw nog eens op een overzichtelijke wijze zien; en omgekeerd laat Figuur 3.2 dus zien hoe dergelijke hiërarchische structuren (zoals bomen en andere grafen) in tabellen kunnen worden weergegeven!



Figuur 3.1: Een grafische weergave van v2(BOM), aangevuld met de omschrijvingen uit v2(PROD)

PNR	OMS	SRT	KPR	VKP	WNST	VRD	SAM
11297	Bout + moer	BM	10	25	15	1024	N
44660	Lade	L	2500	5995	3495	400	J
28325	Ladenhouder	LH	5900	11995	6095	101	N
44704	Lade	L	2500	5995	3495	400	N
87384	Bureau	B	49500	99800	50300	20	J
11198	Duimschroef	AUX	0	100	100	1001	N
11099	Pennenbakje	DIV	50	169	119	144	N
69333	Bureaublad	BB	10000	22500	12500	29	N
81290	Opbergwand	BW	127500	255000	127500	4	J
55803	Ladenkast	K	29000	59800	30800	19	J
52250	Gewone kast	K	25500	59800	34300	37	J
40007	Steun	AUX	500	1295	795	256	J
44066	Plank	DIV	1495	3495	2000	329	N
25762	Ombouw	KO	12890	25780	12890	0	N
40018	Steundeel	AUX	200	550	350	597	N
44022	Ladeplank	LP	300	750	450	70	N
62931	Bureaustoel	ST	4900	9900	5000	0	N

(a) De tabel v2(PROD)

PNR	GW	GWE
62931	9	KG
69333	12	KG
44704	2500	GR
28325	3300	GR
11297	9	GR
44022	178	GR
40018	100	GR
11198	1000	GR
25762	20	KG
44066	1	KG
11099	10	GR

(b) De tabel v2(BAS)

ONR	PNR	AST
28325	87384	2
11297	87384	24
44660	87384	6
11297	44660	6
11198	44660	2
44022	44660	7
69333	87384	1
11198	40007	3
11198	55803	16
11198	52250	16
40007	55803	8
40007	52250	8
40007	81290	16
40018	40007	2
44600	55803	4
44660	81290	3
55803	81290	1
52250	81290	2
44066	52250	4
44066	81290	8
25762	52250	1
25762	55803	1
44704	87384	2

(c) De tabel v2(BOM)

Figuur 3.2: De DB-toestand v2 over GBP

□ Voorbeeld 3.2.

OPGAVE

3.0. Bedenk zelf enige relevante tabelindexen met relevante attributen voor een of andere organisatie (bijvoorbeeld een warenhuis, een ziekenhuis of een software-huis).

□

3.1 SPECIFICATIES VAN TOEGESTANE ATTRIBUUTWAARDEN

Voor elke tabelindex E in dom(g) kunnen de verzamelingen toegestane attribuutwaarden worden vastgelegd door middel van een verzamelingsfunctie F over $g(E)$. Met andere woorden, F is de functie die voor elke attribuut a van E de verzameling toegestane waarden voor a binnen E vastlegt. In [Re 82] heet zo'n verzamelingsfunctie een *objectkarakterisering*.

In de database-literatuur wordt onder een *relatieschema* soms verstaan een verzameling attributen waarbij aan elk attribuut (impliciet) ook een waardenverzameling is "gekoppeld". In dat geval wordt met een relatieschema dus in feite een objectkarakterisering bedoeld. Als voorbeeld van zo'n literatuurplaats noemen we [Mai 83], pagina 2. (Overigens bedoelt men met een relatieschema meestal de verzameling $g(E)$, zoals bijvoorbeeld in [Ul 82] en [Ya 86], dan wel het paar $(E\,;g(E))$, zoals bijvoorbeeld in [Ul 82] en [CP 87].)

Tot nu toe hebben we de volgende voorbeelden van objectkarakteriseringen gezien:

- FM en FA, reeds geïntroduceerd in Voorbeeld 0.1 en gebruikt in Voorbeeld 1.3;
- F4 in Voorbeeld 2.8, gedefinieerd met behulp van de functies F1, F2 en F3 die, hoewel ze wel de vorm van een objectkarakterisering hebben, daar niet als zodanig dienst doen;
- FPCB in Voorbeeld 2.12, ook gedefinieerd met behulp van de "module" F2;
- F5 in Voorbeeld 2.13, gedefinieerd met behulp van S2, de verzameling die de gebruikelijke afkortingen van de twaalf provincies van Nederland opsomt.

In §4.2 is er voor elk van de 19 tabelindexen een objectkarakterisering te vinden. Voorbeelden zijn de functies FPAT, FMW en FMVS. Voorbeelden van attribuutconstraints in Hoofdstuk 4 zijn de randvoorwaarden R03, R04, R05, R06, R46 en R53. Ook R02 is hier een attribuutconstraint. Naast deze "genummerde" attribuutconstraints zijn er ook nog vele niet-genummerde attribuutconstraints; voorbeelden hiervan zijn de eisen dat een intern telefoonnummer 4-cijferig dient te zijn en dat een specialistenhonorarium ten minste 80 gulden per uur moet bijdragen.

Een ander aardig voorbeeld van een attribuutconstraint is de eis dat een bankrekeningnummer altijd aan de zogeheten *11-proef* moet voldoen, dat wil zeggen dat 1 maal het 9^{e} cijfer plus 2 maal het 8^{e} cijfer plus ... plus 8 maal het 2^{e} cijfer plus 9 maal het 1^{e} cijfer een getal moet opleveren dat deelbaar is door 11. De verzameling toegestane bankrekeningnummers is derhalve

$$\{n \in \mathrm{Vng}(9) \mid (\Sigma i \in [1 \,..\, 9] : i * ((n \operatorname{div} 10^{i-1}) \bmod 10)) \bmod 11 = 0\}.$$

Meestal echter is een verzameling toegestane attribuutwaarden van het simpele "type-plus-lengte" formaat, zoals bijvoorbeeld Chs(30), Vng(9) of Int(4). Andere niet ongebruikelijke vormen van verzamelingen toegestane attribuutwaarden zijn intervallen (alias "subranges"), bijvoorbeeld [1 .. 31] voor dagen in een maand of [–273 ..) voor temperaturen.

Soms wordt een verzameling attribuutwaarden bepaald door *enumeratie* (dat wil zeggen door de elementen één voor één op te noemen), zoals bijvoorbeeld de verzameling { ‘MAN’, ‘VROUW’ }, de verzameling { ‘Uitstekend’, ‘Goed’, ‘Voldoende’, ‘Matig’, ‘Slecht’ } of de verzameling {1, 3, 5, 7, 8, 10, 12} (voor de maanden met 31 dagen).

Wanneer het aannemelijk is dat de enumeratieve waardenverzameling van een attribuut *a* in de loop der tijd nog wel eens aan verandering onderhevig zal zijn, dan is het misschien beter om voor *a* een iets ruimere waardenverzameling te kiezen en een aparte "éénkoloms-tabel" te introduceren. Deze nieuwe tabel dient dan op elk moment precies alle op dat moment toegestane waarden voor attribuut *a* te bevatten. Bovendien moet daarbij dan ook de (database)constraint worden toegevoegd dat elke *a*-waarde in de oorspronkelijke tabel ook voorkomt in de zojuist genoemde éénkoloms-tabel; deze databaseconstraint vormt zo als het ware een "variabele attribuutconstraint". Gezien de kennelijke "semantiek" van de drie zojuist genoemde voorbeelden van enumeratieve verzamelingen komt vermoedelijk alleen de "waarderingsverzameling" voor zo’n flexibeler oplossing in aanmerking. In dat geval zou als ruimere waardenverzameling bijvoorbeeld Chs(20) gekozen kunnen worden en de nieuwe tabel zou dan (voorlopig) uit vijf tupels bestaan (zoals weergegeven in Figuur 3.3).

SCHAAL
Uitstekend
Goed
Voldoende
Matig
Slecht

Figuur 3.3

We gaan nu verder met de opbouw van ons BOM-voorbeeld.

VOORBEELD 3.3:

We introduceren als vervolg op Voorbeeld 3.2 één hulpverzameling, te weten de verzameling DPN van mogelijke produktnummers, en drie objectkarakteriseringen:

$$\mathrm{DPN} = \{k \in \mathrm{Vng}(5) \mid k \bmod 11 = 0\};$$

FPROD =	{(PNR	; DPN),	produktnummer	(V01)
	(OMS	; Chs(30)),	omschrijving	
	(SRT	; Chs(6)),	soort produkt	
	(KPR	; $\mathbb{N}$),	kostprijs per stuk	
	(VKP	; $\mathbb{N}$),	verkoopprijs per stuk	
	(WNST	; $\mathbb{N} - \{0\}$),	winst per stuk	(V02)
	(VRD	; $\mathbb{N}$),	voorraad	
	(SAM	; { 'J', 'N' })};	samengesteld produkt?	
FBAS =	{(PNR	; DPN),	produktnummer	
	(GW	; $[1 .. 10^6]$),	gewicht	
	(GWE	; { 'GR', 'KG' })};	gewichtseenheid (gram of kilo)	
FBOM =	{(PNR	; $\{k \in \mathrm{DPN} \mid k \geq 40000\}$),	produktnummer	(V03)
	(ONR	; $\mathbb{N}$),	onderdeelnummer	
	(AST	; [1 ..))}	aantal stuks van dat directe onderdeel	

Merk op dat de als het ware "buiten haakjes" gehaalde verzameling $\{k \in \mathrm{Vng}(5) \mid k \bmod 11 = 0\}$ op drie verschillende plaatsen wordt gebruikt.

Het is eenvoudig na te gaan dat de DB-toestand v2 uit Voorbeeld 3.2 voldoet aan de hierboven gestelde eisen of, om precies te zijn, dat

$$\mathrm{v2(PROD)} \subseteq \Pi(\mathrm{FPROD}) \text{ en } \mathrm{v2(BAS)} \subseteq \Pi(\mathrm{FBAS}) \text{ en } \mathrm{v2(BOM)} \subseteq \Pi(\mathrm{FBOM}).$$

☐ Voorbeeld 3.3.

Voor attribuutwaarden komen niet alleen getallen en strings in aanmerking. In principe kan volgens onze theorie een attribuutwaarde een willekeurig "object" zijn, bijvoorbeeld

- een werktekening (in een CAD/CAM-omgeving bijvoorbeeld),
- een pasfoto (van een werknemer bijvoorbeeld),
- een plattegrond (voor een autonavigatiesysteem bijvoorbeeld),
- een compact disc (in een juke-box bijvoorbeeld),
- een handtekening (van een chef bijvoorbeeld),
- een filmfragment (over een educatief onderwerp bijvoorbeeld),
- een telefoongesprek (met een klant bijvoorbeeld), of
- een röntgenfoto (van een patiënt bijvoorbeeld).

Sommige (zelfs commercieel beschikbare) database-managementsystemen bieden reeds ondersteuning voor zulke "objectgeoriënteerde" mogelijkheden, zie bijvoorbeeld [IDT 88] of [NGI 88]. Een belangrijke vraag hierbij is telkens of het DBMS ook bijpassende operaties op deze bijzondere attribuutwaarden ter beschikking stelt. Voor enkele voorbeelden van objectgeoriënteerde databases en de daarbij behorende operaties verwijzen we naar het overzichtsartikel [HK 87] en naar de daarin genoemde referenties.

Dat onze theorie ook zulke onorthodoxe attribuutwaarden toelaat, is te danken aan het feit dat zij, in tegenstelling tot de meeste andere theorieën, geen a priori veronderstellingen over de aard van de attribuutwaarden bevat. Zo is in het bijzonder de door ons eerder genoemde (doch niet geroemde) eerste-normaalvormeis uit den boze.

OPGAVEN

3.1. Geef de voorwaarden vervat in V01, V02 en V03 van Voorbeeld 3.3 in woorden weer.

3.2. Bedenk voor elk van uw tabelindexen (met bijbehorende attributen) uit Opgave 3.0 een plausibele objectkarakterisering.

□

3.2 SPECIFICATIES VAN TOEGESTANE TUPELWAARDEN

De verzameling toegestane tupels (ofwel tabelementen) voor een tabelindex E met objectkarakterisering F is altijd een deelverzameling van $\Pi(F)$. In de meeste gevallen is het $\Pi(F)$ zelf, zoals tot nu toe in onze voorbeelden ook steeds het geval is geweest, maar in sommige gevallen zijn niet al deze combinaties van attribuutwaarden toegestaan en zal de verzameling toegestane tupels een *echte* deelverzameling van $\Pi(F)$ zijn. Het ontbreken van tupelconstraints wijst er op dat de attributen in kwestie kennelijk "onderling onafhankelijke grootheden" representeren.

In het grote voorbeeld in Hoofdstuk 4 is in negen van de negentien gevallen de verzameling toegestane tupels een echte deelverzameling van het gegeneraliseerde produkt van de betreffende objectkarakterisering. Die echte deelverzamelingen zijn TPAT, TSP, TWN, THA, TPLR, TAFD, TRU, TOPN en TBEH. Deze negen verzamelingen worden bepaald door in totaal vijftien tupelconstraints, te weten de randvoorwaarden R07, R14, R17, R21, R22, R27, R40, R41, R44, R51, R55, R56, R58, R59 en R77.

Andere typische voorbeelden van eisen op tupelniveau, dus eisen *tussen* attribuutwaarden, kunnen zijn: "nettobedrag kleiner dan brutobedrag", "minimaal gewicht (of aantal, of prijs, of tijdsduur) niet groter dan maximaal gewicht (respectievelijk aantal, prijs of tijdsduur)", "(werknemersnummer van) de afdelingschef is niet tevens (dat van) de souschef", "van elke aangesloten vereniging moeten (de lidmaatschapsnummers van) voorzitter, penningmeester en secretaris onderling verschillend zijn" en "als de rekeningsoort niet 'GIRO' is, dan moet het rekeningnummer aan de 11-proef voldoen".

In §4.1 luidt de randvoorwaarde R31 (voor niet-medisch personeel): "Het maandelijkse vergoedingsbedrag mag niet boven het maandsalaris uitkomen". Hoewel R31 op het eerste gezicht een tupelconstraint lijkt, blijkt in §4.2 dat het toch een databaseconstraint is. De reden hiervoor is dat door *generalisatie* - zie §3.4 - de gegevens over het niet-medisch personeel zich in verschillende tabellen bevinden, in casu het vergoedingsbedrag gewoon in de NMP-tabel maar het maandsalaris in de (meer algemene) werknemerstabel. Dit betekent dat R31 moet worden uitgedrukt in termen van de *join* van die twee tabellen. (Binnen dat join-resultaat, wat volgens Lemma 2.5(a) ook een tabel is, is R31 wel een "tupelconstraint".) Eén van de conclusies van dit voorbeeld is dan ook dat de classificatie van een constraint niet een "absolute" eigenschap van die constraint is, maar kan afhangen van het uiteindelijk te kiezen DB-skelet bijvoorbeeld.

VOORBEELD 3.4:

We breiden Voorbeeld 3.3 uit met de volgende verzamelingen toegestane tupels:

TPROD	$= \{t \in \Pi(\text{FPROD}) \mid$	$t(\text{VKP}) \geq 2 * t(\text{KPR})$ en	(V04)
		$t(\text{WNST}) = t(\text{VKP}) - t(\text{KPR})$ en	(V05)
		$(t(\text{VRD}) \leq 2$ of $t(\text{VRD}) * t(\text{KPR}) \leq 10^6)\}$;	(V06)
TBAS	$= \{t \in \Pi(\text{FBAS}) \mid$	als $t(\text{GWE}) =$ 'GR' dan $t(\text{GW}) < 10000\}$;	(V07)
TBOM	$= \{t \in \Pi(\text{FBOM}) \mid$	$t(\text{ONR}) \neq t(\text{PNR})\}$.	(V08)

Merk op dat we nu reeds kunnen constateren dat {VKP, KPR} → {WNST} in TPROD; zie Opgave 3.7.

Omdat FPROD(WNST) = $I\!N - \{0\}$ en FPROD(KPR) = $I\!N$, volgt uit voorwaarde V05 dat bij elementen van TPROD een verkoopprijs nooit 0 is. In feite is dit een voorbeeld van een "verborgen" attribuutconstraint die volgt uit een tupelconstraint.

□ Voorbeeld 3.4.

OPGAVEN

3.3. Geef de vijf voorwaarden in Voorbeeld 3.4 in woorden weer.

3.4. Geef enige elementen van TPROD, van TBAS en van TBOM.

3.5. Geef enige elementen van Π(FPROD) – TPROD, van Π(FBAS) – TBAS en van Π(FBOM) – TBOM.

3.6. Voldoet de DB-toestand v2 uit Voorbeeld 3.2 aan alle in Voorbeeld 3.4 gestelde eisen? (Met andere woorden, is het zo dat v2(PROD) ⊆ TPROD, v2(BAS) ⊆ TBAS en v2(BOM) ⊆ TBOM ?)

3.7. Bewijs het volgende:

(a) {KPR, VKP} → {WNST} in TPROD;

(b) {KPR, WNST} → {VKP} in TPROD;

(c) {VKP, WNST} → {KPR} in TPROD;

3.8. Breid uw oplossing bij Opgave 3.2 uit met bijbehorende verzamelingen toegestane tupels.

□

3.3 SPECIFICATIES VAN TOEGESTANE TABELWAARDEN

Als *Y* de verzameling toegestane tupels voor een tabelindex *E* is, dan is de verzameling toegestane tabellen voor *E* een (meestal echte) deelverzameling *W* van P(*Y*). Dus, elke toegestane tabel voor *E* is een deelverzameling van *Y* terwijl, omgekeerd, niet noodzakelijk elke deelverzameling van *Y* een toegestane tabel voor *E* is.

Tot nu toe hebben we onder andere de volgende voorbeelden van zulke tabellenverzamelingen *W* gezien:

- WM en WA in Voorbeeld 1.3;
- WPCB in Voorbeeld 2.12;
- WP en WG in Voorbeeld 2.13;
- WG2 (en passant) in Voorbeeld 2.14;
- WP2 in Opgave 2.79;
- WPE en WGE in Opgave 2.90.

Verder hebben we in de opgaven 2.47 en 2.101 ook nog enige "kunstmatige" (definities van) tabellenverzamelingen gegeven. De tabellenverzamelingen W1 en W2 in Opgave 2.47 en W3 in Opgave 2.101 zijn namelijk gedefinieerd door één voor één hun *elementen* op te noemen en niet, zoals gebruikelijk, door hun *karakteriserende eigenschappen* op te noemen.

In §4.2 definiëren we 19 tabellenverzamelingen. Ze worden allemaal gebruikt in de definitie van FZKH, vlak voor de definitie van UZKH. Voorbeelden van deze tabellenverzamelingen zijn WPAT, WMW, WMED en WMVS. We tellen daarbij in totaal 33 tabelconstraints.

De belangrijkste eisen op tabelniveau zijn die van *momentane afhankelijkheid* en in het bijzonder die van *unieke identificatie* (zoals ook onze voorbeelden al wel suggereren). Andere mogelijke eisen op tabelniveau zijn bijvoorbeeld eisen aangaande het gemiddelde of totale salaris (al dan niet per afdeling) en eisen omtrent volgnummers, bijvoorbeeld dat ze chronologisch en opeenvolgend zijn toegekend (misschien bij de verschillende medicijnverstrekkingen van eenzelfde patiënt, om maar eens een voorbeeld uit Hoofdstuk 4 te noemen).

Tenslotte noemen we nog als mogelijke eis op tabelniveau dat de waardenverzameling van het ene attribuut in een tabel bevat moet zijn in de waardenverzameling van een zeker ander (meestal uniek identificerend) attribuut van die zelfde tabel. Meer in het algemeen is dit een tabelconstraint van de vorm: *h* verbindt *T* met *T*. Een voorbeeld hiervan is R18 in §4.2. De

randvoorwaarde R26 in §4.2 gaat zelfs nog een stapje verder. Elk van deze eisen is een voorbeeld van een *interne ssr*, waarbij "ssr" een afkorting is voor *subset requirement*. (In de volgende paragraaf zullen we hier nader op ingaan.)

VOORBEELD 3.5:

We breiden nu Voorbeeld 3.4 uit met de volgende verzamelingen toegestane tabellen:

WPROD = $\{T \subseteq$ TPROD $\mid$	{PNR} is u.i. in T en	(V09)
	$(\Sigma\, t \in T : t(\text{VRD}) * t(\text{KPR})) \leq 10^8\}$;	(V10)
WBAS = $\{T \subseteq$ TBAS $\mid$	{PNR} is u.i. in $T\}$;	(V11)
WBOM = $\{T \subseteq$ TBOM $\mid$	{PNR, ONR} is u.i. in T en	(V12)
	$\{(t(\text{ONR}); t(\text{PNR})) \mid t \in T\}$ is acyclisch}.	(V13)

De relatie $\{(t(\text{ONR}); t(\text{PNR})) \mid t \in T\}$ wordt wel een *assemblage-relatie* genoemd en de inverse ervan, dus de relatie $\{(t(\text{PNR}); t(\text{ONR})) \mid t \in T\}$, wel een *explosie-relatie* (in BOM-terminologie).

De eis V13 is een voorbeeld van een zogeheten *recursieve constraint*, en wel omdat er in wezen gebruik wordt gemaakt van *de transitieve afsluiting* van een relatie (zie Definitie 0.22).

Merk op dat V08 nu een "redundante" constraint is geworden; deze tupelconstraint volgt uit de tabelconstraint V13 (aldus Lemma 0.18(d)).

□ Voorbeeld 3.5.

OPGAVEN

3.9. Geef de vijf voorwaarden in Voorbeeld 3.5 in woorden weer.

3.10. Voldoet de DB-toestand v2 uit Voorbeeld 3.2 aan alle in Voorbeeld 3.5 gestelde eisen? (Geef echter, naar analogie van Opgave 3.6, eerst een iets formelere weergave van deze vraag.)

3.11. Geef een element van P(TBOM) – WBOM waarin {PNR, ONR} uniek identificerend is.

3.12. Uit V10 volgt de tupelconstraint dat $t(\text{VRD}) * t(\text{KPR}) \leq 10^8$.

(a) Volgt uit deze tupelconstraint ook V06?

(b) Volgt deze tupelconstraint ook uit V06?

3.13. Ga na of WPROD, WBAS en WBOM in 3NF zijn. Beargumenteer uw antwoorden.

3.14. Breid uw oplossing bij Opgave 3.8 uit met bijbehorende verzamelingen toegestane tabellen.

□

3.4 SPECIFATIES VAN TOEGESTANE DATABASE-WAARDEN

Het uiteindelijke DB-universum over g is een (meestal echte) deelverzameling van $\Pi(H)$ waarbij H de functie over dom(g) is die aan elke E in dom(g) de bij E gekozen verzameling toegestane tabellen toevoegt. Zo'n "tabellenverzamelingswaardige" functie wordt in [Re 82] wel een *database-karakterisering* genoemd.

Tot nu toe hebben we onder andere de volgende voorbeelden van DB-universa (met bijbehorende DB-karakteriseringen) gezien:

- VBU met DB-karakterisering HF in Voorbeeld 1.3;
- U1 met DB-karakterisering F6 in Voorbeeld 2.13;
- U3 met DB-karakterisering F7 in Opgave 2.79.

Verder bevat Opgave 2.91 nog een definitie van een DB-universum in een iets andere gedaante:

$$\text{U4} = \text{U3} \cap \Pi(\text{F8}),$$

waarbij F8 een in Opgave 2.90 gedefinieerde DB-karakterisering is.

Aan het eind van §4.2 vinden we de definitie van het DB-universum UZKH, gebaseerd op de

DB-karakterisering FZKH die aan elk van de 19 tabelindexen één van de eerder gedefinieerde tabellenverzamelingen toevoegt. We tellen 35 databaseconstraints in de definitie van UZKH.

VOORBEELD 3.6:

We ronden de voorbeelden 3.2 tot en met 3.5 af met de introductie van de DB-karakterisering HBP en het DB-universum UBP (over het eerder genoemde DB-skelet GBP). Bovendien voeren we nog de attributentransformatie h5 in; deze functie zullen we gebruiken in voorwaarde V17 (en Opgave 3.17). Volledigheidshalve herhalen we tevens alle voorgaande definities die betrekking hebben op ons BOM-voorbeeld. (Merk echter op dat het DB-skelet GBP verder niet meer expliciet gebruikt hoeft te worden; desalniettemin is het vooraf vaststellen van het onderliggende DB-skelet een zeer belangrijke en moeilijke taak op zichzelf.)

GBP = {(PROD ; {PNR, OMS, SRT, KPR, VKP, WNST, VRD, SAM}),
(BAS ; {PNR, GW, GWE}),
(BOM ; {PNR, ONR, AST})};

h5 = {(ONR	; PNR),	onderdeelnummer
(ONDOMS	; OMS),	onderdeelomschrijving
(SRTOND	; SRT),	soort onderdeel
(KPOND	; KPR),	kostprijs onderdeel
(VKPOND	; VKP),	verkoopprijs onderdeel
(PWOND	; WNST),	potentiële winst op onderdeel
(ONDVRD	; VRD),	voorraadhoogte onderdeel
(ONDSAM	; SAM)};	onderdeel zelf samengesteld?

DPN = $\{k \in \text{Vng}(5) \mid k \bmod 11 = 0\}$;

FPROD = {(PNR	; DPN),	produktnummer	(V01)
(OMS	; Chs(30)),	omschrijving	
(SRT	; Chs(6)),	soort produkt	
(KPR	; $\mathbb{N}$),	kostprijs per stuk	
(VKP	; $\mathbb{N}$),	verkoopprijs per stuk	
(WNST	; $\mathbb{N} - \{0\}$),	winst per stuk	(V02)
(VRD	; $\mathbb{N}$),	voorraad	
(SAM	; {'J', 'N'})};	samengesteld produkt?	

FBAS = $\{$(PNR ; DPN),	produktnummer	
(GW ; $[1 .. 10^6]$),	gewicht	
(GWE ; $\{$‘GR’, ‘KG’$\}$)$\}$;	gewichtseenheid	
FBOM = $\{$(PNR ; $\{k \in \text{DPN} \mid k \geq 40000\}$),	produktnummer	(V03)
(ONR ; $\mathbb{N}$),	onderdeelnummer	
(AST ; $[1 ..)$)$\}$;	aantal stuks	

$$\text{TPROD} = \{t \in \Pi(\text{FPROD}) \mid t(\text{VKP}) \geq 2 * t(\text{KPR}) \text{ en} \qquad (\text{V04})$$
$$t(\text{WNST}) = t(\text{VKP}) - t(\text{KPR}) \text{ en} \qquad (\text{V05})$$
$$(t(\text{VRD}) \leq 2 \text{ of } t(\text{VRD}) * t(\text{KPR}) \leq 10^6)\}; \qquad (\text{V06})$$
$$\text{TBAS} = \{t \in \Pi(\text{FBAS}) \mid \text{als } t(\text{GWE}) = \text{‘GR’ dan } t(\text{GW}) < 10000\}; \qquad (\text{V07})$$
$$\text{TBOM} = \{t \in \Pi(\text{FBOM}) \mid t(\text{ONR}) \neq t(\text{PNR})\}; \qquad (\text{V08})$$

$$\text{WPROD} = \{T \subseteq \text{TPROD} \mid \{\text{PNR}\} \text{ is u.i. in } T \text{ en} \qquad (\text{V09})$$
$$(\Sigma t \in T : t(\text{VRD}) * t(\text{KPR})) \leq 10^8\}; \qquad (\text{V10})$$
$$\text{WBAS} = \{T \subseteq \text{TBAS} \mid \{\text{PNR}\} \text{ is u.i. in } T\}; \qquad (\text{V11})$$
$$\text{WBOM} = \{T \subseteq \text{TBOM} \mid \{\text{PNR, ONR}\} \text{ is u.i. in } T \text{ en} \qquad (\text{V12})$$
$$\{(t(\text{ONR}); t(\text{PNR})) \mid t \in T\} \text{ is acyclisch}\}; \qquad (\text{V13})$$

HBP = $\{$(PROD ; WPROD),	produkten
(BAS ; WBAS),	basisprodukten
(BOM ; WBOM)$\}$;	stuklijstregels

$$\text{UBP} = \{v \mid v \in \Pi(\text{HBP}) \text{ en}$$
$$\{(\text{ONR; PNR})\} \text{ verbindt } v(\text{BOM}) \text{ met } v(\text{PROD}) \text{ en} \qquad (\text{V14})$$
$$\text{id}(\{\text{PNR}\}) \text{ verbindt } v(\text{BOM}) \text{ bilateraal}$$
$$\text{met } \{t \in v(\text{PROD}) \mid t(\text{SAM}) = \text{‘J’}\} \text{ en} \qquad (\text{V15})$$
$$\text{id}(\{\text{PNR}\}) \text{ verbindt } v(\text{BAS}) \text{ bilateraal}$$
$$\text{met } \{t \in v(\text{PROD}) \mid t(\text{SAM}) = \text{‘N’}\} \text{ en} \qquad (\text{V16})$$
$$\forall p \in v(\text{PROD}):$$
$$(\Sigma t \in v(\text{BOM}) \bowtie (v(\text{PROD}) \infty \text{ h5}) \text{ en } t(\text{PNR}) = p(\text{PNR}):$$
$$t(\text{KPOND}) * t(\text{AST})) \leq p(\text{KPR})\}. \qquad (\text{V17})$$

De laatste eis geeft weer dat de kostprijs van een produkt minimaal de som van de kostprijzen van de (eventuele) directe onderdelen is.

Uit voorwaarde V14 volgt, in combinatie met V01, dat alle ONR-waarden in de BOM-tabel elementen van DPN zijn. We zien hier dus een voorbeeld van een "verborgen" attribuutconstraint die volgt uit een databaseconstraint. Enerzijds hadden we deze (voor de hand liggende) attribuutconstraint meteen kunnen opnemen in de betreffende objectkarakterisering, anderzijds blijkt deze eis dus in principe overbodig te zijn.

□ Voorbeeld 3.6.

Veruit de belangrijkste klasse van databaseconstraints is de algemene klasse van *verbindingseisen*, ook wel *subset requirements* (of kortweg ssr's) genoemd. Verbindingseisen zijn er echter in allerlei "soorten en maten". We zullen daarom een paar speciale deelklassen nader bekijken.

(A1) De volgende (elementaire) vorm van verbindingseisen komt in de praktijk het meest voor:

h verbindt $v(M)$ met $v(D)$

Deze "standaardvorm" wordt ook wel een *referential integrity constraint* genoemd. Als bovendien rng(h) een sleutel van de D-tabel is, wat meestal wel het geval is, dan wordt dom(h) ook wel een *foreign key* genoemd. Een voorbeeld van zo'n elementaire verbindingseis waarbij deze situatie zich voordoet, is V14 in Voorbeeld 3.6. De attributenverzameling {ONR} van BOM is dus een foreign key volgens dit woordgebruik.

In de definitie van UZKH in §4.2 zijn de eerste 20 databaseconstraints die daar genoemd worden van bovenstaande vorm. Strikt genomen is er alleen bij de formulering van de randvoorwaarde R52 niet sprake van een foreign key. (Wel is er een binnen UZKH equivalente herformulering van R52 mogelijk waarbij {SCNR} wel een foreign key blijkt.)

(A2) Een sterkere maar minder vaak gebruikte vorm van verbindingseis is:

h verbindt $v(M)$ bilateraal met $v(D)$

Het klassieke voorbeeld is hier dat van de orders en de orderregels. Het ordernummer van elke orderregel moet ook voorkomen in de ordertabel, terwijl omgekeerd elke order ten minste één orderregel moet bevatten (zodat dus het ordernummer van elke order ook moet voorkomen in de orderregeltabel).

Als h injectief is, dan is een eis van de vorm (A2) volgens Lemma 2.15 ook te

schrijven als twee afzonderlijke eisen van de vorm (A1).

Als rng(h) een sleutel van de D-tabel en dom(h) een sleutel van de M-tabel is, dan is in elke toestand v de door h geïnduceerde associatie volgens Lemma 2.16 in feite een bijectie van $v(M)$ op $v(D)$. We hadden bij de modellering in fase 0 de tabelindexen M en D dan ook net zo goed kunnen samennemen tot één tabelindex die zowel de attributen van M als die van D erft (minus dom(h), omdat die attributenverzameling al is vertegenwoordigd door rng(h)).

(B1) Soms is het zo dat elk M-tupel moet corresponderen met een D-tupel dat voldoet aan een speciale nevenvoorwaarde:

h verbindt $v(M)$ met $\{t \in v(D) \mid \phi(t)\}$

Anders gezegd, de eis $\phi(t)$ is een noodzakelijke voorwaarde voor een D-tupel t om bijbehorende M-tupels te kunnen hebben. Deze vorm is dus een versterking van de vorm onder (A1). Voorbeelden van databaseconstraints van deze vorm zijn R65, R74 en R84 in §4.2 en de onder (A1) genoemde equivalente herformulering van R52.

(B2) De volgende vorm is een versterking van (B1):

h verbindt $v(M)$ bilateraal met $\{t \in v(D) \mid \phi(t)\}$

Hier is het dus zo dat de nevenvoorwaarde *precies* karakteriseert bij welke D-tupels er ook nog M-tupels bestaan. Voorbeelden van randvoorwaarden van deze vorm zijn V15 en V16 in Voorbeeld 3.6 en R16, R19, R28, R30 en R67 in §4.2.

Zoals de zojuist aangehaalde voorbeelden misschien al deden vermoeden, is de eis $\phi(t)$ vaak van de vorm "$t(a) = w_0$", waarbij de waarde w_0 niet afhangt van het tupel t of de DB-toestand v. In dat geval luidt de databaseconstraint dus:

h verbindt $v(M)$ bilateraal met $\{t \in v(D) \mid t(a) = w_0\}$

We noemen in dit geval het attribuut a wel een *inspectie-attribuut van D* en w_0 wel de *inspectie-waarde voor M*. Ook hier is het dus zo dat men "in de D-tabel zelf" precies kan zien bij welke D-tupels er nog M-tupels behoren, en wel aan de a-waarde van de D-tupels.

Uit V15 en V16 in Voorbeeld 3.6 blijkt dat SAM een inspectie-attribuut van PROD is. De inspectie-waarde voor BOM is 'J' en voor BAS is het 'N'. Verder blijkt uit R16,

R28 en R30 in §4.2 dat SRTM een inspectie-attribuut van MW is. De inspectie-waarden voor de tabelindexen SP, MP en NMP zijn hier "toevalligerwijze" achtereenvolgens 'SP', 'MP' en 'NMP'. Volgens R67 in §4.2 is ONTS een inspectie-attribuut van OPN met 'JA' als de inspectie-waarde voor de tabelindex ONT.

Dat de zojuist gegeven bloemlezing van klassen van verbindingseisen zeker geen volledige opsomming van alle mogelijkheden is, moge blijken uit de verbindingseisen R24, R08 en R37 in §4.2.

Als in één van de voorgaande verbindingseisen M en D toevallig dezelfde tabelindexen zijn (en in geval van een B1- of B2-vorm de nevenvoorwaarde ϕ ook niet over andere tabelindexen rept), dan noemen we de betreffende ssr wel een *interne ssr* en is die constraint in feite een *tabel*constraint. (Zo reduceert in dit geval bijvoorbeeld de vorm (A1) tot de vorm: h verbindt T met T.) In §3.3 noemden we als voorbeelden reeds de interne ssr's R18 en R26 uit §4.2. De tabelconstraint R18 is hierbij van de vorm behandeld onder (A1) en R26 is van de algemene vorm behandeld onder (B1).

Als in één van de voorgaande verbindingseisen de tabelindexen verschillend zijn (of de nevenvoorwaarde ϕ wel over andere tabelindexen rept), dan noemen we de betreffende ssr een *externe ssr* en is er kennelijk sprake van een echte databaseconstraint.

Een verbindingseis "komt zelden alleen" (zoals we al eerder lieten doorschemeren): Zo'n databaseconstraint treedt meestal op in combinatie met de tabelconstraint dat rng(h) uniek identificerend in $v(D)$ is. In dat geval is volgens Lemma 2.16(3) de door h geïnduceerde associatie op $v(M) \times v(D)$ een functie. Eén van de weinige tegenvoorbeelden is hier de onder (A2) genoemde verbindingseis dat elk ordernummer in de ordertabel ook in de orderregeltabel moet voorkomen.

Als, gegeven een verbindingseis van één van de eerder genoemde vormen, niet alleen rng(h) een sleutel van de D-tabel maar ook dom(h) een sleutel van de M-tabel is, dan is volgens Lemma 2.16 in elke toestand de door h geïnduceerde associatie een injectieve functie van $v(M)$ naar $v(D)$. Anders gezegd, met elk D-tupel correspondeert er maximaal één M-tupel. Het is nu dus in elke DB-toestand zo dat er voor sómmige D-tupels als het ware nog enige additionele attributen van toepassing zijn, te weten de elementen van $g(M)$ – dom(h), waarbij $g(M)$ de attributenverzameling van de tabelindex M volgens het DB-skelet g is.

De hierboven geschetste situatie komt onder andere voor rondom de verbindingseis V16 in Voorbeeld 3.6, waar {PNR} zowel een sleutel van de BAS-tabel als van de PROD-tabel is. Dus voor sommige PROD-tupels zijn er nog de additionele attributen GW en GWE (gewicht

en gewichtseenheid) uit de BAS-tabel van toepassing. Merk op dat deze (niet zo verrassende) *formele conclusie* conform de laatste opmerking in de vorige alinea in ieder geval in overeenstemming is met de oorspronkelijke *informele omschrijving* verwoord in Voorbeeld 3.2. Andere voorbeelden van verbindingseisen waarbij de bovengenoemde situatie zich voordoet zijn R33, R16, R19, R28, R30, R67 en R49 in §4.2 en de laatste eis in de definitie van U1 in Voorbeeld 2.13. Wanneer we nu de laatste opmerking in de vorige alinea toepassen op de verbindingseis uit Voorbeeld 2.13, dan komen we tot het (wel verrassende) inzicht dat woonplaatsattributen ook kunnen worden opgevat als burgemeestersattributen. Toepassing op de verbindingseis R49 in §4.2 leidt nu tot de conclusie dat afdelingsattributen ook kunnen worden opgevat als souschef-attributen in dat voorbeeld. Dat het *afdelings*nummer uiteindelijk toch "wezenlijker" voor een AFD-tupel is dan het *souschef*nummer zal pas blijken uit de dynamische constraints in §5.2, in het bijzonder uit C00. (Zie ook de opmerkingen over primary keys en alternate keys in §5.1.)

Als de door h geïnduceerde associatie in elke toestand v van een DB-universum U een injectieve functie van $v(M)$ naar $v(D)$ is, dan zeggen we wel dat M een *specialisatie* of *differentiatie* van D is. In sommige gevallen wordt M ook wel een *subtype* van D genoemd. Volgens deze terminologie is bijvoorbeeld BAS een specialisatie van PROD in Voorbeeld 3.6 (dankzij V16), VV een specialisatie van WN in §4.2 (dankzij R33) en ONT een specialisatie van OPN in §4.2 (dankzij R67).

We noemen een tabelindex D een *generalisatie* van een tabelindexverzameling X in een DB-universum U als elk element M van X een specialisatie van D is en er bovendien in elke toestand van U bij elk D-tupel precies één tupel van precies één tabelindex uit X behoort. Bij de drie zojuist genoemde specialisaties is er geen sprake van generalisatie. Wel is in §4.2 de tabelindex MW een generalisatie van {SP, WN} en bovendien is WN zelf weer een generalisatie van {MP, NMP}.

We willen er nog op wijzen dat er bij bilaterale verbindingseisen sprake is van een zogenaamd *kip/ei-probleem*: Een nieuwe dom(h)-waarde bestemd voor de M-tabel vraagt ook om een nieuwe rng(h)-waarde voor de D-tabel en . . . omgekeerd. Dit betekent dus dat er bij overgang naar een andere DB-toestand "in één keer" zowel een M-tupel als een D-tupel moet worden toegevoegd! Meer in het algemeen doet zich bij een DB-universum U over een DB-skelet g een kip/ei-situatie voor wanneer de volgende relatie cyclisch is:

$$\{(M\,;D) \in \mathrm{dom}(g) \times \mathrm{dom}(g) \mid \exists\, h \in g(M) \,\tilde{\twoheadrightarrow}\, g(D) : h \text{ verbindt } M \text{ permanent met } D \text{ in } U\},$$

of, in woorden, de verzameling van alle indexparen $(M\,;D)$ waarvoor geldt dat M permanent wordt verbonden met D in U door een of andere attributentransformatie h.

OPGAVEN

3.15. Geef de voorwaarden V14, V15 en V16 uit Voorbeeld 3.6 in woorden weer.

3.16. Is de DB-toestand v2 uit Voorbeeld 3.2 een element van UBP uit Voorbeeld 3.6?

3.17. Geef voor $v \in$ UBP in het kort de verschillen tussen de volgende tabellen weer.

(a) v(BOM) $\bowtie$ v(PROD);

(b) v(BOM) $\bowtie$ (v(PROD) ∞ h5);

(c) v(BOM) $\bowtie$ v(PROD) $\bowtie$ (v(PROD) ∞ h5).

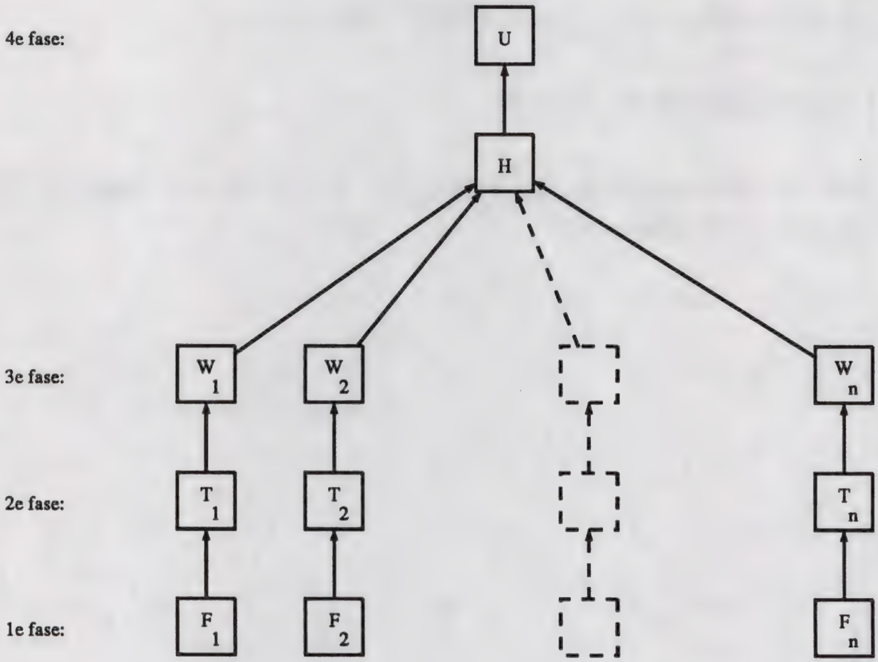
3.18. Leg uit waarom UBP niet in 3NV is.

3.19. Rond uw oplossingen bij de opgaven 3.0, 3.2, 3.8 en 3.14 af met de definitie van een bijbehorend DB-universum.

□

3.5 SAMENVATTING

We hebben in de constructie van een DB-universum over een gegeven DB-skelet vier fasen onderscheiden. In elke fase worden er één of meer waardenverzamelingen gedefinieerd. We kunnen de onderlinge samenhang tussen die afzonderlijke "modules" in deze gefaseerde opbouw van de definitie van een DB-universum U als volgt schematisch weergeven:



waarbij $T_i \subseteq \Pi(F_i)$ en

$W_i \subseteq \mathbf{P}(T_i)$ en

$H = \{(E_1 ; W_1), (E_2 ; W_2), \ldots, (E_n ; W_n)\}$ en

$U \subseteq \Pi(H)$ en

waarbij $\boxed{A} \rightarrow \boxed{B}$ betekent: A wordt gebruikt in de definitie van B.

Mogelijk worden hierbij de definities van de objectkarakteriseringen F_1 tot en met F_n nog voorafgegaan door de definities van enige hulpbegrippen (bijvoorbeeld constanten, verzamelingen of functies), zoals onder andere in de voorbeelden 2.13 en 3.6 en in §4.2. In database-kringen worden de hulpverzamelingen die als waardenverzamelingen voor

attributen dienst doen (helaas) ook wel "domeinen" genoemd. Dit kan echter verwarring geven met de klassieke betekenis van het woord "domein", dat is het domein van een functie, of van een relatie in het algemeen (zie Hoofdstuk 0).

We merken verder op dat deze hiërarchie van definities in (ten minste) twee verschillende volgorden is te presenteren:

(A) Per *tabelindex* werkend, dat wil zeggen per tabelindex de resultaten van de fasen 1, 2 en 3 direct achter elkaar presenteren. (Patroon: $(F\,;T\,;W)^*$.)

(B) Per *fase* werkend, dat wil zeggen eerst alle definities uit fase 1 presenteren, dan alle definities uit fase 2 en dan alle definities uit fase 3. (Patroon: $F^*\,;T^*\,;W^*$.)

Beiden gevallen eindigen met de definities van de DB-karakterisering H en (last but not least) het DB-universum U.

We hebben volgorde (A) gebruikt in Voorbeeld 2.13 en §4.2. Volgorde (B) hebben we gebruikt in Voorbeeld 1.3, in het overzicht aan het eind van §2.4 (voor in wezen dezelfde definities als in Voorbeeld 2.13) en in het huidige hoofdstuk. In Hoofdstuk 8 zullen we de definities zelfs een paar keer "van achteren naar voren" presenteren (wat in dit verband ook wel *top-down* heet): we geven daar soms eerst de definitie van het DB-universum, daarna de definitie van de daarin genoemde database-karakterisering, daarna de definities van de daarin genoemde tabellenverzamelingen, enzovoort.

Soortgelijke opmerkingen betreffende presentatievolgorden zijn overigens van toepassing op de te definiëren hulpbegrippen: in Voorbeeld 2.13 en in §2.4 staat elk hulpbegrip in feite direct vóór de (eerste) objectkarakterisering waarin het wordt gebruikt, terwijl in Voorbeeld 3.6 en in §4.2 alle hulpbegrippen helemaal in het begin staan.

We zullen laten zien dat de hierboven beschreven constructie inderdaad altijd een DB-universum oplevert, en wel over het DB-skelet g gedefinieerd door
$g = \{(E_1\,;\mathrm{dom}(F_1)), (E_2\,;\mathrm{dom}(F_2)), \ldots, (E_n\,;\mathrm{dom}(F_n))\}$:

> We beginnen met de opmerking dat F_i een verzamelingsfunctie is.
> Daarom is niet alleen $\Pi(F_i)$ maar ook elke deelverzameling T_i van $\Pi(F_i)$ een tabel over $\mathrm{dom}(F_i)$, en dus over $g(E_i)$.
> Daarmee is niet alleen $\mathbf{P}(T_i)$ maar ook elke deelverzameling W_i van $\mathbf{P}(T_i)$ een verzameling tabellen over $g(E_i)$.
> Nu volgt uit de eerder gegeven beschrijving van H dat elk element v van $\Pi(H)$ een

functie over dom(g) is, en wel zodanig dat $v(E_i)$ voor elke E_i in dom(g) een element van W_i is.
Uit de vorige twee conclusies volgt nu dat $v(E_i)$ een tabel over $g(E_i)$ is.
Volgens Definitie 1.3 is nu $\Pi(H)$, en daarmee ook elke deelverzameling U van $\Pi(H)$, een DB-universum over g.

Hiermee hebben we aangetoond dat op bovenstaande manier altijd een DB-universum wordt verkregen. (Dit geldt trouwens ook bij willekeurig veel tabelindexen en niet alleen bij een eindig aantal, zoals hierboven gesuggereerd.) In Opgave 3.22 wordt (impliciet) gevraagd aan te tonen dat, omgekeerd, elk DB-universum op deze manier kan worden beschreven.

Een vraag die men zich na het definiëren van een DB-universum zeker zou moeten stellen, is de vraag of het zojuist gedefinieerde DB-universum wel *consistent* is, dat wil zeggen of het eigenlijk wel elementen bevat, zie Opgave 1.5. Immers, het zou kunnen dat de - in de praktijk vaak omvangrijke - definitie interne conflicten bevat zonder dat men zich dat meteen bewust is. Bovenstaande vraag zou (positief) beantwoord kunnen worden door een voorbeeld van een toestand te geven en aan te tonen dat het voorbeeld inderdaad aan alle in de definitie gestelde voorwaarden voldoet.

Een toestand die men meestal als element van het te definiëren DB-universum over g wenst, is "de lege toestand" over de betreffende verzameling tabelindexen, dat wil zeggen de toestand $\lambda E \in \mathrm{dom}(g) : \varnothing$; deze toestand kan namelijk goed dienen als "starttoestand" (bij initialisatie). Als men aantoont dat de lege toestand een element van het zojuist gedefinieerde DB-universum is, dan heeft men daarmee uiteraard ook de in de vorige alinea genoemde vraag meteen beantwoord (en wel in positieve zin).

Als men weet dat een DB-universum U consistent is, dan rest nog de vraag of elke tabelindex E van U wel *consistent in U* is, dat wil zeggen of de E-tabel überhaupt wel ooit elementen kan bevatten. (Volgens de in Opgave 1.5 geïntroduceerde terminologie is dit dan in feite de vraag of *U regulier* is.) Gewoonlijk kan deze vraag in positieve zin worden beantwoord door één toestand te geven waarin elke tabel niet-leeg is, en weer aan te tonen dat het voorbeeld inderdaad aan alle in de definitie gestelde voorwaarden voldoet. (Dat dit wel voldoende maar niet altijd mogelijk is voor reguliere DB-universa blijkt uit één van de opgaven in Hoofdstuk 1.)

OPGAVEN

3.20. Geef een "zo klein mogelijk" element van UBP uit Voorbeeld 3.6 waarin elke tabel niet-leeg is.

3.21. Laat U een DB-universum over een verzamelingsfunctie g zijn.

(a) Is er voor U een "kleinste" database-karakterisering?
Zo ja, welke is het en in welke zin is de database-karakterisering het kleinst?

(b) Is er voor elke tabelindex E van U een "kleinste" karakteriserende tupelverzameling? Zo ja, definieer dan deze verzameling tupels.

(c) Is er voor elke tabelindex E van U een "kleinste" objectkarakterisering?
Zo ja, welke is het en in welke zin is die objectkarakterisering het kleinst?

3.22. Als U een DB-universum over een verzamelingsfunctie g is, dan is er voor elke tabelindex E van U een verzamelingsfunctie F_E zodanig dat $U \subseteq \Pi(\lambda E \in \mathrm{He}(U) : \mathrm{P}(\Pi(F_E)))$.
Bewijs dit.

□

4 EEN NIET-TRIVIAAL VOORBEELD VAN EEN DATABASE-UNIVERSUM

In dit hoofdstuk presenteren we een niet-triviaal voorbeeld van een DB-universum om enig inzicht te geven in de praktische problemen die zich kunnen voordoen bij ontwerp en gebruik van *complexe* databases (dat wil zeggen databases met veel tabelindexen, zeg enige tientallen, veel attributen, zeg enige honderden in totaal, veel constraints, en veel relevante DB-functies, zeg ook enige tientallen). Naast *massa* (dat wil zeggen zeer veel tupels in de actuele toestand, zeg enige tienduizenden of enige miljoenen) is complexiteit een van de belangrijke bronnen van problemen bij databases zoals die in de praktijk voorkomen.

Ons voorbeeld gaat over een (denkbeeldig) streekziekenhuis en is een variant op de voorbeelden in [Re 82] en [Br 84], onder meer uitgebreid met enige ideeën afkomstig uit [GOV 84]. Er zijn 19 tabelindexen en in totaal 128 attributen. Verder zijn er in het voorbeeld vele soorten constraints (waaronder diverse zogeheten "business rules") verwerkt, afkomstig uit zeer uiteenlopende toepassingsgebieden (maar allemaal "vertaald" naar deze ene ziekenhuistoepassing). Ook zijn er enige tientallen interessante DB-functies.

De overwegingen bij het invullen van de details van het voorbeeld waren vooral van didactische en niet zozeer van medisch-organisatorische aard. Er is naar gestreefd de meeste soorten statische constraints die we in de praktijk zijn tegengekomen binnen één voorbeeld te laten zien, en niet om een voor ziekenhuizen zo geschikt mogelijke database-structuur te ontwerpen.

Ook is er niet naar gestreefd het voorbeeld in andere (ontwerp)opzichten "optimaal" te maken; zo is het DB-universum bijvoorbeeld niet in de derde normaalvorm. Op deze wijze kan het voorbeeld tevens dienen om de consequenties van bepaalde ontwerpbeslissingen, bijvoorbeeld aanwezigheid van redundante gegevens, in de context van een complexe database te laten zien.

Het hier geïntroduceerde voorbeeld is ook reeds enige malen gebruikt als "functional benchmark" om bestaande database-managementsystemen op hun datadefinitiemogelijkheden te testen (zie [NGI 87], [NGI 88] en [IDT 88]).

Van ons ziekenhuisvoorbeeld geven we in §4.1 een informele beschrijving en in §4.2 de formele definitie. Om het verband tussen de beide paragrafen gemakkelijk te kunnen aangeven hebben we sommige attribuutconstraints en *alle* tupel-, tabel- en database-constraints genummerd (R01, R02, etcetera). De informele beschrijvingen zijn overigens niet altijd voldoende scherp geformuleerd om daar de uiteindelijke formele definitie zonder meer uit te destilleren. Dit is echter representatief voor de gang van zaken in de praktijk. Vaak zal men dan ook nog eens naar de betreffende "materiedeskundige" terug moeten gaan om te vragen hoe het nu precies zit met die constraint.

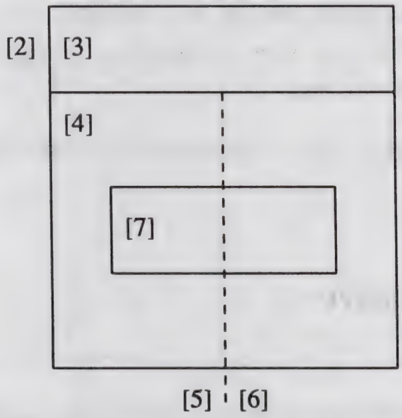
Tot besluit geven we in §4.3 enige voorbeelden van DB-functies over het in §4.2 gedefinieerde DB-universum.

4.1 INFORMELE BESCHRIJVING

We beginnen de informele beschrijving van ons ziekenhuisvoorbeeld met een opsomming van de gebruikte tabelindexen en per tabelindex een korte omschrijving van de objecten die door de tupels van de bijbehorende tabel worden gerepresenteerd:

[1] PAT : ingeschreven patiënten
[2] MW : medewerkers (generalisatie van [3] en [4])
[3] SP : specialisten (ook vroegere)
[4] WN : werknemers (generalisatie van [5] en [6])
[5] MP : medisch personeel
[6] NMP : niet-medisch personeel
[7] VV : verplicht verzekerde werknemers (differentiatie van [4])
[8] HA : huisartsen in de regio
[9] PLR : plaatsen in de regio
[10] AFD : afdelingen
[11] RU : ruimten (ook vroegere en geplande)
[12] OPN : opnamen (ook vroegere)
[13] ONT : ontslagen
[14] OSP : overdrachten (ook vroegere)
[15] OVR : overplaatsingen (ook vroegere)
[16] BEH : behandelingen
[17] PB : (patiënt)behandelingen (ook vroegere)
[18] MED : medicijnen
[19] MVS : medicijnverstrekkingen (ook vroegere)

De bovengenoemde generalisaties geven een onderverdeling van medewerkers in specialisten en werknemers en vervolgens van werknemers in medisch en niet-medisch personeel. Verder wordt met bovengenoemde differentiatie aangegeven dat sommige werknemers verplicht verzekerden zijn. Een en ander wordt schematisch weergegeven in Figuur 4.1.



$[2] = [3] \cup [4]$ en $[3] \cap [4] = \varnothing$

$[4] = [5] \cup [6]$ en $[5] \cap [6] = \varnothing$

$[7] \subseteq [4]$

Figuur 4.1: Generalisatie en differentiatie

We merken op dat deze gelijkheden en inclusies alleen gelden voor de betreffende verzamelingen te representeren *objecten* maar niet, zoals straks zal blijken, voor de betreffende verzamelingen *tupels*!

In elk van de volgende 19 alinea's wordt telkens één van de tabellen nader toegelicht.

Van alle **ingeschreven patiënten** moeten de volgende gegevens worden bijgehouden: patiëntnummer, NAW-gegevens (dat wil zeggen naam, adres, postcode en woonplaats), geboorteplaats, geboortedatum, datum van inschrijving, bloedgroep, rhesusfactor, geslacht, (registratie)nummer van de huisarts, naam en praktijkadres van de huisarts en van de tandarts, naam en adres van de apotheek, verder van elk der kinderen voornaam (roepnaam), geboortedatum en geslacht, en ten slotte een rubriek "diversen" voor allerlei andere wetenswaardigheden. Iedere ingeschreven patiënt heeft zijn (of haar) eigen patiëntnummer [R01] en per patiënt moet ieder kind een eigen R01
voornaam hebben [R02]. Een patiëntnummer moet een 6-cijferig getal zijn [R03]. R02,R
Elk patiëntnummer moet bovendien deelbaar zijn door 9 [R04]. Het eerste cijfer van R04
elk patiëntnummer kan op deze wijze worden opgevat als een controlecijfer, dit met het oog op eventuele fouten. (Ook in de praktijk komen dergelijke *foutendetecterende codes* voor maar ze zijn dan vaak subtieler van aard; een bekend voorbeeld is de 11-proef voor bankgironummers.) Ons ziekenhuis begon te draaien op 1 april 1960 en

daarom zal elke datum die betrekking heeft op een ziekenhuisactiviteit (waaronder ook inschrijfdatum) na 31 maart 1960 liggen [R05]. Toen het ziekenhuis in 1960 begon, kon een geboortedatum teruggaan tot 1850 [R06]. We merken verder op dat de inschrijfdatum niet vóór de geboortedatum van een ingeschreven patiënt mag liggen [R07]. Bovendien moet elke ingeschreven patiënt in de regio wonen [R08]. De betreffende huisarts zal een zogeheten "regio-arts" moeten zijn [R09]. (Het begrip regio-arts zal verderop worden uitgelegd.) R05 R06 R07,R08 R09

De volgende relevante gegevens zijn van toepassing op *elke* **medewerker**: identiteitsnummer (een getal van 3 of 4 cijfers), NAW-gegevens, afdelingsnummer, intern telefoonnummer (4-cijferig), soort medewerker (specialist, medisch personeel of niet-medisch personeel) en een rubriek "diversen". Het identiteitsnummer bepaalt de medewerker eenduidig [R10]. Elke medewerker is formeel verbonden aan een afdeling van het ziekenhuis [R11]. Medewerkers met hetzelfde interne telefoonnummer moeten op dezelfde afdeling werken [R12]. R10 R11 R12

Naast de algemene medewerkersgegevens zijn de volgende additionele kenmerken specifiek van toepassing op onze **specialisten**: identiteitsnummer van de zogeheten locum tenens (plaatsvervanger), soort dienstverband, specialismen (elke specialist moet één of meer specialismen hebben), telefoonnummer thuis, contractueel aantal uren per week (met een maximum van 50 uur), honorarium per uur (met een wettelijk minimum van 80 gulden) en aantal toegezegde bedden; ook moet worden bijgehouden of de specialist in kwestie nog steeds bij het ziekenhuis werkzaam is of niet. Dit laatste is nodig omdat we niet alleen de huidige specialisten bijhouden, maar ook alle vroegere specialisten (*cumulatief* dus). Als nadere toelichting op het aantal toegezegde bedden merken we op dat elke specialist op papier recht heeft op een bepaald aantal verpleegbedden; per specialist wordt dit aantal contractueel vastgelegd. (In het ziekenhuis spreken we in dit verband ook wel van "papieren bedden" omdat er geen "fysieke bedden" worden gereserveerd.) Het totale aantal toegezegde verpleegbedden mag echter het aantal daadwerkelijk aanwezige verpleegbedden met niet meer dan 10% overschrijden [R13]. Voor specialisten die niet meer bij het ziekenhuis werkzaam zijn, stellen we het aantal toegezegde bedden op nul, evenals het aantal uren per week [R14]. Voor specialisten bestaat het identiteitsnummer uit drie cijfers. Elke specialist heeft, zijnde een medewerker, zijn eigen identiteitsnummer [R15]. Voor elke medewerker geeft het kenmerk "soort medewerker" precies aan of die medewerker een specialist is of niet [R16]. De plaatsvervanger van een specialist moet een andere [R17], (ooit) aan het ziekenhuis verbonden specialist zijn [R18]. (De voor de hand liggende eis dat de plaatsvervanger van een nog in dienst zijnde specialist ook nog in dienst moet zijn, is de informatie-analisten van het ziekenhuis R13 R14 R15 R16 R17,R18

kennelijk ontgaan; een andere reden kan zijn dat men de eis niet belangrijk genoeg vond om op te nemen.)
We merken op dat de eisen R15 en R16 expliciet gesteld moeten worden om (samen met de onderstaande eisen R25 en R19) de eerstgenoemde generalisatie aan het begin van §4.1 formeel gestalte te kunnen geven.

Onder een **werknemer** wordt (per definitie) verstaan een medewerker die niet als spe-
cialist geboekt staat [R19]. Voor deze categorie medewerkers moeten de volgende R19
extra gegevens worden bijgehouden: identiteitsnummer van de groepsleider, aantal
halve (werk)dagen per week (omdat deelbetrekkingen mogelijk zijn), maandsalaris en
aantal ziektedagen tot nu toe in het lopende jaar. Wegens de weekends en de feest-
dagen kan het aantal ziektedagen (eigenlijk: verzuimde werkdagen) nooit meer dan
290 bedragen. Het salaris kan hooguit gelijk zijn aan het ambtelijke maximumsalaris.
Het ambtelijke maximumsalaris bedraagt momenteel 10720 gulden per maand [R20]. R20
Bij een (deel)betrekking kan het salaris hooguit (een evenredig deel van) het amb-
telijke maximumsalaris bedragen [R21]; het salaris mag echter niet beneden (het R21
evenredige deel van) het ambtelijke minimumsalaris liggen [R22]. Het ambtelijke R22
minimumsalaris bedraagt momenteel 975 gulden per maand [R23]. De voormalige R23
werknemers worden niet bijgehouden; dus, anders dan bij specialisten, staan bij
werknemers alleen de huidige geregistreerd. Alle werknemers moeten in de regio
wonen [R24]. Voor werknemers bestaat het identiteitsnummer uit vier cijfers. Elke R24
werknemer heeft, een medewerker zijnde, zijn eigen identiteitsnummer [R25]. Elke R25
groepsleider is een werknemer die bovendien zichzelf formeel als groepsleider heeft
[R26]. Groepsleiders mogen geen deeltijdbaan hebben [R27]. R26,R2

Tot het **medische personeel** behoren (per definitie) die medewerkers die als zodanig
staan geboekt [R28]. (Specialisten worden dus blijkbaar niet tot het (gewone) R28
medische personeel gerekend; tot het medische personeel kunnen alleen maar werkne-
mers behoren.) Voor het medische personeel zijn naast de medewerkers- en werkne-
mersgegevens ook het telefoonnummer thuis en de code van de (hoogste) vooroplei-
ding van belang. Elk medisch personeelslid heeft, een medewerker zijnde, zijn eigen
identiteitsnummer [R29]. R29

Tot het **niet-medische personeel** behoren per definitie die medewerkers die als zodanig staan geboekt [R30]. Voor het niet-medische personeel zijn naast de R30
medewerkers- en werknemersgegevens ook de volgende drie gegevens relevant: functiecode, maandelijkse vergoedingsbedrag - hoewel een niet-medisch personeelslid diverse toeslagen en andere extra vergoedingen kan krijgen, houden we hier alleen het totale maandbedrag bij - en ten slotte het aantal vrije dagen per jaar waarop het betreffende personeelslid recht heeft op basis van een volledige betrekking. Het maandelijkse vergoedingsbedrag mag niet boven het maandsalaris uitkomen [R31]. Elk R31
niet-medisch personeelslid heeft, zijnde een medewerker, zijn eigen identiteitsnummer
[R32]. De eisen R32 en R30 dienen samen met de eisen R29 en R28 als expliciete R32
weergave van de tweede generalisatie die aan het begin van §4.1 werd genoemd.

Van alle **verplicht verzekerde werknemers** moeten de volgende extra gegevens worden bijgehouden: ziekenfondsnummer (8-cijferig), registratienummer van de huisarts en naam en adres van de apotheek. Elke genoemde verplicht verzekerde moet ook als werknemer geregistreerd staan [R33]. Daar elke werknemer in de regio moet wonen, R33
zal ook de betreffende huisarts een regio-arts moeten zijn [R34]. Iedere verplicht R34
verzekerde werknemer heeft niet alleen zijn eigen identiteitsnummer [R35] maar ook R35
zijn eigen ziekenfondsnummer [R36]. De eisen R33 en R35 zijn nodig om de aan het R36
begin van §4.1 genoemde differentiatie formeel gestalte te geven.

In verband met bepaalde regionale (controle)taken die ons ziekenhuis heeft te vervullen, is het noodzakelijk dat van alle **huisartsen met een praktijk in de regio**, de zogeheten "regio-artsen", de volgende gegevens worden bijgehouden: registratienummer, naam, praktijkadres - moet dus in de regio zijn [R37] - , telefoonnummer thuis en R37
op de praktijk, spreekuurgegevens, aantal patiënten op 1 januari jongstleden (de vaste jaarlijkse peildatum), toename van het aantal patiënten in het afgelopen jaar en verwachte toename voor het lopende jaar. (Eventuele afname wordt genoteerd als negatieve toename.) Het registratienummer bepaalt de huisarts eenduidig [R38], R38
evenals de combinatie van naam en praktijkadres [R39]. Uiteraard kan de toename R39
van het aantal patiënten in het afgelopen jaar niet groter zijn dan het aantal patiënten
op 1 januari jl. [R40]. Evenmin kan de eventueel verwachte afname in het lopende R40
jaar groter zijn dan het aantal patiënten op 1 januari jl. [R41]. R41

Eveneens in het kader van de door ons (streek)ziekenhuis te vervullen regionale controletaken dienen van alle **plaatsen in de regio** enige gegevens te worden bijgehouden, te weten de plaatsnaam (die binnen de regio uniek moet zijn [R42]), het R42
inwoneraantal (in duizendtallen), het aantal huisartsen, apotheken en tandartsen werkzaam in die plaats, naam en (praktijk)adres van de huisarts en van de apotheek

die voor die plaats het aanstaande weekend de weekenddienst waarnemen, en ten slot-
te het identiteitsnummer van de aan ons ziekenhuis verbonden contactpersoon voor
die plaats. Een contactpersoon is altijd een niet-medisch personeelslid [R43], meestal R43
een maatschappelijk werker die de lokale situatie en problematiek goed kent. Voor
plaatsen waarin ten minste vier huisartsen werkzaam zijn, moet de weekenddienst
altijd worden waargenomen door een huisarts wiens praktijk gevestigd is in de betref-
fende plaats zelf [R44]. In ieder geval moet de dienstdoende huisarts er een zijn met R44
een praktijk in de regio [R45]. R45

Van alle **afdelingen** van ons ziekenhuis moeten worden bijgehouden het afdelings-
nummer, de afdelingsnaam, het identiteitsnummer van de chef en van de souschef - op
sommige afdelingen ook wel hoofd en subhoofd genoemd - en de soort; er worden
hierbij vijf soorten onderscheiden: medisch, administratief, laboratorium, dienst
(waaronder bijvoorbeeld de centrale technische dienst en de dienst bouwzaken) en
"diversen" [R46]. Elke afdeling heeft haar eigen nummer [R47] en haar eigen naam R46,R4
[R48]. De souschef van een afdeling moet een medewerker van die afdeling zijn R48
[R49]. De chef van een afdeling moet ook een medewerker zijn [R50], maar niet R49,R5
noodzakelijk van dezelfde afdeling. (Zodoende kan iemand in principe chef van meer
afdelingen zijn.) In geen enkele afdeling mogen de rollen van chef en souschef in
dezelfde persoon zijn verenigd [R51]. Het is echter wel mogelijk dat iemand tegelijk R51
souschef van de ene afdeling en chef van een andere afdeling is. Voor de rol van
souschef komen alleen groepsleiders in aanmerking [R52]. R52

Van alle actuele, vroegere en bij bestaande afdelingen nu reeds geplande **ruimten**
moeten gegevens worden bijgehouden, en wel de volgende: ruimtenummer (bestaande
uit een code van maximaal negen tekens), afdelingsnummer, vloeroppervlakte, status
(aanwezig, gepland of vervallen), soort ruimte en aantal aanwezige bedden. Er worden
vijf soorten ruimten onderscheiden: behandelruimten, verpleegruimten, recepties (en
balies), magazijnen en "diversen" [R53]. Alle actuele, vroegere en geplande ruimten R53
binnen ons ziekenhuis hebben hun eigen ruimtenummer [R54]. (De nummers van R54
reeds vervallen ruimten mogen dus niet opnieuw worden gebruikt.) In een ruimte
mogen maximaal 15 bedden staan. In elke ruimte moet er per bed ten minste vier vier-
kante meter aan vloeroppervlakte zijn [R55]. In aanwezige behandelruimten en R55
aanwezige verpleegruimten kunnen bedden staan, in andere ruimten echter niet [R56]. R56
Daar afdelingen bij ons nooit vervallen, zullen alle aanwezige, vroegere en bij be-
staande afdelingen reeds geplande ruimten altijd tot bestaande afdelingen moeten
behoren [R57]. R57

Van alle lopende en reeds beëindigde **opnamen** moeten gegevens worden bijgehouden. Onder een opname wordt hier verstaan een aaneengesloten verblijfperiode van een patiënt in het ziekenhuis. De volgende kenmerken zijn relevant voor *elke* opname: het nummer van de opgenomen patiënt, de datum waarop de opname begon, het aanvankelijke ruimtenummer, het identiteitsnummer van de aanvankelijk verantwoordelijke persoon, de aanvankelijke opnamereden, de naam en het praktijkadres van de huisarts op moment van opname, de dieetgegevens, de aanvankelijke verpleegkundige bevindingen en de ontslagdatum (bij lopende opnamen de *geplande* vertrekdatum en bij reeds beëindigde opnamen de *feitelijke* vertrekdatum). Bij ruimtenummer en verpleegkundige bevindingen bedoelen we met *aanvankelijk* "tot eventuele overplaatsing naar een andere ruimte", en bij verantwoordelijke persoon en opnamereden bedoelen we met *aanvankelijk* "tot eventuele overdracht van verantwoordelijkheid". Toelichting: Een patiënt kan gedurende een opname wel eens naar een andere ruimte worden overgeplaatst en, in principe onafhankelijk daarvan, ook wel eens onder de verantwoordelijkheid van iemand anders gaan vallen; hierop komen we later terug.

Als afronding van onze opsomming van de relevante opnamegegevens merken we op dat bij elke opname ook nog moet worden aangegeven of die opname reeds is afgelopen en of de patiënt tijdens die opname is overleden. Als een patiënt overlijdt dan
wordt de opname als beëindigd beschouwd [R58]. Opnamen duren in principe ten R58
minste één nacht, maar overlijdensgevallen kunnen een uitzondering op deze regel
vormen [R59]. Een patiënt kan per dag maar één keer worden opgenomen [R60]. R59,R60
Verschillende opnamen van een zelfde patiënt overlappen elkaar niet (hoewel de
ontslagdatum van een opname wel de begindatum van de volgende opname kan zijn)
[R61]. Per patiënt kan er ten hoogste één lopende opname zijn waarvoor dan alleen de R61
opname met de meest recente begindatum in aanmerking komt [R62]. Een opname R62
kan uiteraard alleen gevolgd worden door een andere opname van dezelfde patiënt als
de betreffende patiënt tijdens de eerstgenoemde opname niet is overleden [R63]. Een R63
opgenomen patiënt moet ingeschreven staan [R64] en een opname kan alleen maar R64
plaatsvinden (of hebben plaatsgevonden) in één van de aanwezige of vroegere *verpleeg*ruimten [R65]. De aanvankelijk verantwoordelijke voor een opname moet altijd R65
een van onze specialisten zijn [R66]. R66

Voor elke **beëindigde opname** zijn er twee extra kenmerken van belang, namelijk het factuurbedrag en de nazorginstructies. (Bij elk ontslag uit het ziekenhuis wordt er een factuur - ofwel rekening - opgemaakt, waarvan we hier alleen het eindbedrag bijhouden.) Voor een opname moet altijd minimaal 240 gulden in rekening worden gebracht. De genoemde extra kenmerken zijn voor elke beëindigde opname van toepassing en voor geen enkele lopende opname [R67]. Voor een zelfde patiënt R67
kunnen er verscheidene beëindigde opnamen zijn, maar patiëntnummer plus opnamedatum te zamen bepalen elke beëindigde opname eenduidig [R68]. Wat de sugges- R68
tieve naamgeving al deed vermoeden is dankzij de eisen R67 en R68 nu ook expliciet weergegeven: de verzameling beëindigde opnamen vormt een differentiatie van de verzameling van alle opnamen.

Zoals reeds eerder is opgemerkt, is het mogelijk dat gedurende een opname de verantwoordelijkheid voor die opname wordt overgedragen aan een andere specialist. Meestal gebeurt dit in verband met een nieuwe (opname)reden. We willen dan ook voor elke **overdracht**, dat wil zeggen elke verandering van verantwoordelijke specialist, het nummer van de patiënt in kwestie, de datum van overdracht, het identiteitsnummer van de nieuwe specialist en de (eventueel nieuwe) opnamereden bijhouden. Een overdracht kan, administratief gezien, per patiënt ten hoogste één keer per dag plaatsvinden [R69]; de reden hiervoor is dat er toch maar één verandering relevant is R69
wanneer een patiënt op een zelfde dag meer dan eens van verantwoordelijke specialist verandert. (Dit zal meestal de laatste verandering op die dag zijn.) Een overdracht kan bovendien alleen maar plaatsvinden *tijdens* een opname (van de betreffende patiënt) die op dat moment ten minste één nacht heeft geduurd [R70]. Evenals bij een opname R70
moet ook bij een overdracht de nieuwe verantwoordelijke altijd een van onze specialisten zijn [R71]. Het is mogelijk dat een patiënt gedurende een opname wordt R71
overgeplaatst naar een andere ruimte. We willen van elke **overplaatsing** bijhouden het nummer van de patiënt die verhuist, de datum van overplaatsing, het nummer van de nieuwe ruimte en de nieuwe verpleegkundige bevindingen tijdens het verblijf in die nieuwe ruimte. Evenals een verandering van specialist kan, administratief gezien, een overplaatsing per patiënt ten hoogste één keer per dag plaatsvinden [R72] en wel R72
alleen maar *tijdens* een opname (van de betreffende patiënt) die op dat moment ten minste één nacht heeft geduurd [R73]. Een overplaatsing kan alleen maar plaats- R73
vinden (of hebben plaatsgevonden) naar één van de aanwezige of vroegere *verpleeg*ruimten [R74]. R74

Van alle **behandelingen** die we kunnen uitvoeren, moeten de volgende gegevens worden bijgehouden: code, naam, soort, tarief, minimaal en maximaal verwachte duur, en ten slotte het geschatte aantal keren dat de behandeling in kwestie in het lopende jaar

zal worden uitgevoerd; deze prognose moet in het bijzonder een specificatie per
maand geven. Zowel de code [R75] als de naam [R76] bepaalt de behandeling R75,R76
eenduidig. De maximaal verwachte duur mag uiteraard niet kleiner zijn dan de
minimaal verwachte duur, maar eventueel wel gelijk daaraan [R77]. R77

Van elke **uitvoering van een behandeling** houden we bij: het nummer van de behandelde patiënt, de code van de uitgevoerde behandeling, het identiteitsnummer van de behandelende specialist (ook wel hoofdbehandelaar genoemd), het identiteitsnummer van de eventuele assistent-specialist (als er bij de behandeling geen assistent-specialist is, dan noteren we als assistent-specialist de hoofdbehandelaar zelf), het ruimtenummer, de behandelingsdatum, de feitelijke duur, het factuurbedrag en een volgnummer.
Dit volgnummer wordt gebruikt om de verschillende toepassingen van een zelfde
behandeling op dezelfde patiënt van elkaar te onderscheiden [R78]. Per patiënt R78
worden de verschillende toepassingen van dezelfde behandeling opeenvolgend en
chronologisch genummerd, te beginnen bij 1 [R79]. De hoofdbehandelaar moet altijd R79
één van onze eigen specialisten zijn (geweest) [R80], en hetzelfde geldt voor de R80
assistent-specialist [R81]. Verder moet de behandelde patiënt staan (of meteen R81
worden) ingeschreven [R82] en tevens moet de uitgevoerde behandeling in ons R82
"assortiment" voorkomen [R83]. Ten slotte merken we op dat een patiëntbehandeling R83
alleen maar kan plaatsvinden (of hebben plaatsgevonden) in één van de aanwezige of
vroegere *behandel*ruimten [R84]. R84

Ten behoeve van onze medische staf (dat wil zeggen specialisten plus medisch personeel) en onze eigen apotheek (zie de volgende alinea) moeten van diverse **medicijnen** enige gegevens worden bijgehouden, te weten de medicijncode, de naam, de soort, de gevarencode (die kan lopen van 1 t/m 20), de mogelijke bijwerkingen, het aantal eenheden in voorraad, de te berekenen prijs per eenheid en de voor die medicijn gebruikelijke eenheid. Afhankelijk van het geneesmiddel wordt de eenheid aangeduid met GR voor gram, MG voor milligram, ML voor milliliter - waarvoor echter bij sommige medicijnen nog altijd CC wordt gebruikt - of ST voor stuks. Elk geneesmiddel
heeft zowel zijn eigen medicijncode [R85] als zijn eigen naam [R86]. R85,R86

Ons ziekenhuis heeft een eigen apotheek die, op voorschrift van een specialist, medicijnen aan patiënten kan verstrekken. Van alle **medicijnverstrekkingen** tot nu toe moeten de volgende gegevens worden bijgehouden: patiëntnummer, medicijncode, datum van ingang, identiteitsnummer van de voorschrijvende specialist, voorgeschreven duur (met een maximum van 90 dagen), voorgeschreven aantal keren per dag (maximaal 6), voorgeschreven aantal eenheden per keer, factuurbedrag (minimaal een rijksdaalder) en een volgnummer. Dit volgnummer wordt gebruikt om de

verschillende medicijnverstrekkingen aan een zelfde patiënt van elkaar te onder-
scheiden [R87]. Per patiënt worden de verschillende medicijnverstrekkingen R87
opeenvolgend en chronologisch genummerd, te beginnen bij 1 [R88]. De voor- R88
schrijvende specialist moet één van onze eigen specialisten zijn (geweest) [R89], de R89
patiënt moet zijn ingeschreven [R90] en het voorgeschreven geneesmiddel moet bij R90
onze apotheek bekend zijn [R91]. Tot besluit noemen we de randvoorwaarde dat onze R91
ziekenhuisapotheek aan geen enkele patiënt hetzelfde geneesmiddel meer dan eens per
dag mag verstrekken [R92]. R92

De formele definities geven we in §4.2. Die paragraaf heeft de volgende (modulaire) opbouw:

- De eerste pagina begint met de definities van enige "hulpconstanten" en "hulpdomeinen" (MINSAL tot en met KNDVZ).
- Daarna volgen er voor elk van de 19 "entiteiten":
  - Een *objectkarakterisering* (FPAT, FMW, FSP, . . . , FMED, FMVS); hier komen we alleen *attribuutconstraints* tegen.
  - Eventueel nog een expliciet genoemde verzameling toegestane tupels (TPAT, TSP, . . . , TOPN, TBEH), namelijk voor die (negen) entiteiten waarvoor er *tupelconstraints* zijn.
  - De verzameling toegestane tabellen voor die entiteit (WPAT, WMW, WSP, . . . , WMED, WMVS); hierin zijn de *tabelconstraints* verwerkt.
- De voorlaatste pagina bevat de *database-karakterisering* FZKH, die de 19 tabelindexen introduceert en daaraan de bijbehorende verzameling toegestane tabellen toevoegt.
- Last but not least bevat de laatste pagina de definitie van ons *DB-universum* UZKH; hier treffen we alle *databaseconstraints* aan.

4.2 FORMELE DEFINITIE

Definities		*Toelichting*
MINSAL	= 975;	zie R23
MAXSAL	= 10720;	zie R20
GBDVZ	= [18500101 .. 19991231];	zie R06
DATVZ	= [600401 .. 991231];	zie R05
RMSVZ	= { 'BHR' , 'VPR' , 'RCPT' , 'MAG' , 'DIV' };	zie R53
SRTVZ	= { 'MED' , 'ADM' , 'LAB' , 'DNST' , 'DIV' };	zie R46
PNRVZ	= $\{k \mid k \in$ Vng(6) en	zie R03
	k mod 9 = 0};	zie R04
TELVZ	= Π({(NETN; [10 .. 9999]),	netnummer (zonder de nul)
	(ABNR ; [100 .. 9999999])});	abonneenummer
NAWVZ	= Π({(NM ; Chs(20)),	naam
	(ADR ; Chs(20)),	adres
	(PC ; Chs(7)),	postcode
	(WPL ; Chs(15))});	(woon)plaats
KTUVZ	= Π({(VNM ; Chs(14)),	voornaam
	(GBD ; GBDVZ),	geboortedatum
	(GESL ; { 'M' , 'V' }))});	geslacht
KNDVZ	= $\{T \mid T \subseteq$ KTUVZ en	
	{VNM} is u.i. in T};	zie R02
		INGESCHREVEN PATIËNTEN:
FPAT	= {(PNR ; PNRVZ),	patiëntnummer
	(NAW ; NAWVZ),	NAW-gegevens
	(GPL ; Chs(15)),	geboorteplaats
	(GBD ; GBDVZ),	geboortedatum
	(INSD ; DATVZ),	inschrijfdatum
	(BLGR ; { 'O' , 'A' , 'B' , 'AB' }),	bloedgroep
	(RHF ; { '+' , '−' }),	rhesusfactor
	(GESL ; { 'M' , 'V' }),	geslacht
	(KNDG ; KNDVZ),	gegevens der kinderen
	(NRHA ; [1 .. 9999]),	nummer van de huisarts
	(NAWH ; NAWVZ),	naam en praktijkadres huisarts
	(NAWT ; NAWVZ),	naam en praktijkadres tandarts
	(NAWA ; NAWVZ),	naam en adres apotheek
	(DIV ; Chs(800))};	diversen
TPAT	= $\{t \mid t \in \Pi$ (FPAT) en	
	t(GBD) $\leq$ t(INSD) + 19000000};	zie R07
WPAT	= $\{T \mid T \subseteq$ TPAT en	
	{PNR} is u.i. in T};	zie R01

	MEDEWERKERS:
FMW = {(IDNR ; Vng(3) $\cup$ Vng(4)),	identiteitsnummer
(NAW ; NAWVZ),	NAW-gegevens
(ANR ; [1 .. 100]),	afdelingsnummer
(TELI ; Vng(4)),	telefoonnummer intern
(SRTM ; { 'SP' , 'MP' , 'NMP' }),	soort medewerker
(DIV ; Chs(1000))};	diversen
WMW = {T | $T \subseteq \Pi$(FMW) en	
{IDNR} is u.i. in T en	zie R10
{TELI} $\rightarrow$ {ANR} in T};	zie R12
	SPECIALISTEN:
FSP = {(IDNR ; Vng(3)),	identiteitsnummer
(LOC ; Vng(3)),	id. nr. van de plaatsvervanger
(SDV ; Chs(15)),	soort dienstverband
(SPEC ; $\mathbf{P}$(Chs(15)) – {$\varnothing$}),	specialismen
(TELT ; TELVZ),	telefoonnummer thuis
(HONU ; [8000 ..)),	honorarium per uur (in centen)
(AUW ; [0 .. 50]),	aantal uren per week
(ABD ; $\mathbb{N}$),	aantal toegezegde (verpleeg)bedden
(IND ; { 'JA' , 'NEE' })};	nog in dienst?
TSP = {t | $t \in \Pi$ (FSP) en	
t (LOC) $\neq$ t (IDNR) en	zie R17
als t (IND) = 'NEE'	zie R14
dan (t (ABD) = 0 en t (AUW) = 0)};	"
WSP = {T | $T \subseteq$ TSP en	
{IDNR} is u.i. in T en	zie R15
{(LOC; IDNR)} verbindt T met T};	zie R18
	WERKNEMERS:
FWN = {(IDNR ; Vng(4)),	identiteitsnummer
(GLNR ; Vng(4)),	id. nr. van de groepsleider
(AHD ; [1 .. 10]),	aantal halve dagen per week
(SAL ; [0 .. MAXSAL]),	maandsalaris (in guldens)
(AZD ; [0 .. 290])};	aantal ziektedagen in het lopende jaar
TWN = {t | $t \in \Pi$ (FWN) en	
t (SAL) $\geq$ (MINSAL * t (AHD)) div 10 en	zie R22
t (SAL) $\leq$ (MAXSAL * t (AHD)) div 10 en	zie R21
als t (IDNR) = t (GLNR) dan t (AHD) = 10};	zie R27
WWN = {T | $T \subseteq$ TWN en	
{IDNR} is u.i. in T en	zie R25
{(GLNR; IDNR)} verbindt T	zie R26
met {$t \in T$ | t (GLNR) = t (IDNR)}};	"

	MEDISCH PERSONEEL:
FMP = {(IDNR ; Vng(4)),	identiteitsnummer
(TELT ; TELVZ),	telefoonnummer thuis
(OPLC ; Chs(8))};	vooropleidingscode
WMP = $\{T \mid T \subseteq \Pi$ (FMP) en	
{IDNR} is u.i. in $T\}$;	zie R29
	NIET-MEDISCH PERSONEEL:
FNMP = {(IDNR ; Vng(4)),	identiteitsnummer
(FCD ; Chs(10)),	functiecode
(AVD ; [15 .. 30]),	aantal vrije dagen per jaar
(VERG ; $\mathbb{N}$)};	vergoedingsbedrag (in guldens)
WNMP = $\{T \mid T \subseteq \Pi$ (FNMP) en	
{IDNR} is u.i. in $T\}$;	zie R32
	VERPLICHT VERZEKERDEN:
FVV = {(IDNR ; Vng(4)),	identiteitsnummer
(ZFNR ; Vng(8)),	ziekenfondsnummer
(NRHA ; [1 .. 9999]),	nummer van de huisarts
(NAWA; NAWVZ)};	naam en adres apotheek
WVV = $\{T \mid T \subseteq \Pi$ (FVV) en	
{IDNR} is u.i. in T en	zie R35
{ZFNR} is u.i. in $T\}$;	zie R36
	HUISARTSEN IN DE REGIO:
FHA = {(NRHA ; [1..9999]),	nummer van de huisarts
(NAWH; NAWVZ),	naam en praktijkadres
(TELP ; TELVZ),	telefoonnummer praktijk
(TELT ; TELVZ),	telefoonnummer thuis
(SPRU ; Chs(25)),	spreekuur
(PAT ; $\mathbb{N}$),	aantal patiënten op 1 januari jl.
(TOEN ; $\mathbb{Z}$),	patiëntentoename vorig jaar
(VWTN ; $\mathbb{Z}$)};	verwachte toename dit jaar
THA = $\{t \mid t \in \Pi$ (FHA) en	
t(PAT) - t (TOEN) ≥ 0 en	zie R40
t(PAT) + t (VWTN) $\geq 0\}$;	zie R41
WHA = $\{T \mid T \subseteq$ THA en	
{NRHA} is u.i. in T en	zie R38
{NAWH} is u.i. in $T\}$;	zie R39

	PLAATSEN IN DE REGIO:
FPLR = {(WPL ; Chs(15)),	plaatsnaam
(INWA ; [0 .. 1000]),	inwoneraantal (in duizendtallen)
(CPRS ; Vng(4)),	id. nr. contactpersoon
(DDHA; NAWVZ),	dienstdoende huisarts a.s. weekend
(DDAP; NAWVZ),	dienstdoende apotheek a.s. weekend
(AHA ; $\mathbb{N}$),	aantal huisartsen werkzaam in die plaats
(AAP ; $\mathbb{N}$),	aantal apotheken in die plaats
(ATA ; $\mathbb{N}$)};	aantal tandartsen werkzaam in die plaats
TPLR = {t | $t \in \Pi$ (FPLR) en	
als t (AHA) $\geq$ 4	zie R44
dan t (DDHA) (WPL) = t (WPL)};	„
WPLR = {T | $T \subseteq$ TPLR en	
{WPL} is u.i. in T};	zie R42
	AFDELINGEN:
FAFD = {(ANR ; [1 .. 100]),	afdelingsnummer
(ANM ; Chs(20)),	afdelingsnaam
(CHNR; Vng(4) $\cup$ Vng(3)),	id. nr. van de chef
(SCNR ; Vng(4)),	id. nr. van de souschef
(SRT ; SRTVZ)};	soort afdeling
TAFD = {t | $t \in \Pi$ (FAFD) en t(CHNR) $\neq$ t (SCNR)};	zie R51
WAFD = {T | $T \subseteq$ TAFD en	
{ANR} is u.i. in T en	zie R47
{ANM} is u.i. in T};	zie R48
	RUIMTEN:
FRU = {(RNR ; Chs(9)),	ruimtenummer
(ANR ; [1 .. 100]),	afdelingsnummer
(OPVL ; $\mathbb{N}$),	oppervlakte (in m^2)
(STAT ; { 'AANW' , 'GEPL' , 'VERV' }),	status
(SRTR ; RMSVZ),	soort ruimte
(ABD ; [0 .. 15])};	aantal aanwezige bedden
TRU = {t | $t \in \Pi$ (FRU) en	
t (OPVL) $\geq$ 4 * t (ABD) en	zie R55
als t (ABD) $\neq$ 0	zie R56
dan t (STAT) = 'AANW' en	„
t (SRTR) $\in$ { 'BHR' , 'VPR' }};	„
WRU = {T | $T \subseteq$ TRU en	
{RNR} is u.i. in T};	zie R54

	OPNAMEN:
FOPN = {(PNR ; PNRVZ),	patiëntnummer
(OPND ; DATVZ),	opnamedatum
(RNR ; Chs(9)),	verpleegruimtenummer
(IDNR ; Vng(3)),	id. nr. verantwoordel. specialist
(OPNR ; Chs(40)),	opnamereden
(NAWH; NAWVZ),	toenmalige huisarts
(DGEG ; Chs(240)),	dieetgegevens
(BEV ; Chs(1200)),	verpleegkundige bevindingen
(ONTD ; DATVZ),	(geplande) ontslagdatum
(ONTS ; { 'JA' , 'NEE' }),	ontslagen?
(OVRL ; { 'JA' , 'NEE' })};	overleden?
TOPN = {$t \mid t \in \Pi$ (FOPN) en t(OPND) $\leq$ t(ONTD) en	zie R59
(als t(OPND) = t(ONTD) dan t(OVRL) = 'JA') en	"
(als t(OVRL) = 'JA' dan t(ONTS) = 'JA')};	zie R58
WOPN = {$T \mid T \subseteq$ TOPN en	
{ PNR, OPND} is u.i. in T en	zie R60
$\forall t \in T : \forall t' \in T$: als t (PNR) = t'(PNR) en	
t (OPND) < t'(OPND)	
dan t (ONTD) $\leq$ t' (OPND) en	zie R61
t (ONTS) = 'JA' en	zie R62
t (OVRL) = 'NEE' };	zie R63
	ONTSLAGEN:
FONT = {(PNR ; PNRVZ),	patiëntnummer
(OPND ; DATVZ),	opnamedatum
(BEDR ; [24000 ..)),	factuurbedrag (in centen)
(NZRG ; Chs(160))};	nazorginstructies
WONT = {$T \mid T \subseteq \Pi$ (FONT) en	
{PNR, OPND} is u.i. in T};	zie R68
	OVERDRACHTEN:
FOSP = {(PNR ; PNRVZ),	patiëntnummer
(BDAT ; DATVZ),	datum van overdracht
(IDNR ; Vng(3)),	id. nr. nieuwe specialist
(OPNR ; Chs(40))};	(nieuwe) opnamereden
WOSP = {$T \mid T \subseteq \Pi$ (FOSP) en	
{ PNR, BDAT} is u.i. in T};	zie R69

	OVERPLAATSINGEN:
FOVR = {(PNR ; PNRVZ),	patiëntnummer
(BDAT ; DATVZ),	datum van overplaatsing
(RNR ; Chs(9)),	nieuw verpleegruimtenummer
(BEV ; Chs(400))};	additionele bevindingen
WOVR = {$T \mid T \subseteq \Pi$ (FOVR) en	
{ PNR, BDAT} is u.i. in T};	zie R72
	BEHANDELINGEN:
FBEH = {(BCD ; Chs(4)),	behandelcode
(BNM ; Chs(50)),	behandelnaam
(BSRT ; Chs(3)),	behandelsoort
(TAR ; [10 ..)),	tarief (in guldens)
(MIND ; [1 ..)),	min. verwachte duur (in minuten)
(MAXD; [1 ..)),	max. verwachte duur (in minuten)
(PRGN ; [1 .. 12] $\rightarrow \mathbb{N}$)};	jaarprognose maandelijks aanbod
TBEH = {$t \mid t \in \Pi$ (FBEH) en t (MIND) $\leq$ t(MAXD)};	zie R77
WBEH = {$T \mid T \subseteq$ TBEH en	
{BCD} is u.i. in T en	zie R75
{BNM} is u.i. in T};	zie R76
	PATIËNTBEHANDELINGEN:
FPB = {(PNR ; PNRVZ),	patiëntnummer
(BCD ; Chs(4)),	behandelcode
(VNR ; [1 ..)),	volgnummer
(IDNR ; Vng(3)),	id. nr. behandelende specialist
(ASNR ; Vng(3)),	id. nr. assistent-specialist
(RNR ; Chs(9)),	nr. van de behandelruimte
(BDAT ; DATVZ),	behandelingsdatum
(DUUR; [1 ..)),	duur (in minuten)
(BEDR ; [1000 ..))};	factuurbedrag (in centen)
WPB = {$T \mid T \subseteq \Pi$ (FPB) en	
{ PNR, BCD, VNR} is u.i. in T en	zie R78
$\forall t \in T$: als t(VNR) $\neq 1$	zie R79
dan $\exists t' \in T : t' \upharpoonright$ {PNR, BCD} = $t \upharpoonright$ {PNR, BCD}	"
en t' (VNR) = t (VNR) – 1	"
en t' (BDAT) $\leq$ t (BDAT)};	"

	MEDICIJNEN:
FMED = {(MCD ; Chs(6)),	medicijncode
(MNM ; Chs(20)),	medicijnnaam
(MSRT ; Chs(4)),	medicijnsoort
(GVC ; [1 .. 20]),	gevarencode
(BIJW ; **P** (Chs(15))),	mogelijke bijwerkingen
(VRD ; $\mathbb{N}$),	aantal eenheden in voorraad
(EENH ; { 'GR', 'MG', 'ML', 'CC', 'ST' }),	eenheid
(VKPR ; $\mathbb{N}$)};	prijs per eenheid (in centen)
WMED = {$T \mid T \subseteq \Pi$ (FMED) en	
{MCD} is u.i. in T en	zie R85
{MNM} is u.i. in T};	zie R86
	MEDICIJNVERSTREKKINGEN:
FMVS = {(PNR ; PNRVZ),	patiëntnummer
(MCD ; Chs(6)),	medicijncode
(DAT ; DATVZ),	datum van ingang
(VNR ; [1 ..)),	volgnummer
(IDNR ; Vng(3)),	id. nr. voorschrijvende specialist
(DUUR ; [1 .. 90]),	voorgeschreven duur (in dagen)
(FREQ ; [1 .. 6]),	frequentie per dag
(AEK ; [1 ..)),	aantal eenheden per keer
(BEDR ; [250 ..))};	factuurbedrag (in centen)
WMVS = {$T \mid T \in \Pi$ (FMVS) en	
{ PNR, MCD, DAT} is u.i. in T en	zie R92
{ PNR, VNR} is u.i. in T en	zie R87
$\forall t \in T$: als t (VNR) $\neq 1$	zie R88
dan $\exists t' \in T$: t' (PNR) = t (PNR) en	,,
t' (VNR) = t (VNR) $-$ 1 en	,,
t' (DAT) $\leq t$ (DAT)};	,,

FZKH = {(PAT ; WPAT),	ingeschreven patiënten
(MW ; WMW),	medewerkers
(SP ; WSP),	specialisten (ook vroegere)
(WN ; WWN),	werknemers
(MP ; WMP),	medisch personeel
(NMP ; WNMP),	niet-medisch personeel
(VV ; WVV),	verplicht verzekerde werknemers
(HA ; WHA),	huisartsen in de regio
(PLR ; WPLR),	plaatsen in de regio
(AFD ; WAFD),	afdelingen
(RU ; WRU),	ruimten (ook vroegere en geplande)
(OPN ; WOPN),	opnamen (ook vroegere)
(ONT ; WONT),	ontslagen
(OSP ; WOSP),	overdrachten (ook vroegere)
(OVR ; WOVR),	overplaatsingen (ook vroegere)
(BEH ; WBEH),	behandelingen
(PB ; WPB),	patiëntbehandelingen (ook vroegere)
(MED ; WMED),	medicijnen
(MVS ; WMVS)};	medicijnverstrekkingen (ook vroegere)

UZKH = $\{v \mid v \in \Pi$ (FZKH) en	
id({PNR}) verbindt v(OPN) met v(PAT) en	zie R64
id({PNR}) verbindt v(PB) met v(PAT) en	zie R82
id({PNR}) verbindt v(MVS) met v(PAT) en	zie R90
{(SCNR; IDNR), (ANR; ANR)} verbindt v(AFD) met v(MW) en	zie R49
{(CHNR; IDNR)} verbindt v(AFD) met v(MW) en	zie R50
{(ASNR; IDNR)} verbindt v(PB) met v(SP) en	zie R81
id({IDNR}) verbindt v(PB) met v(SP) en	zie R80
id({IDNR}) verbindt v(OPN) met v(SP) en	zie R66
id({IDNR}) verbindt v(OSP) met v(SP) en	zie R71
id({IDNR}) verbindt v(MVS) met v(SP) en	zie R89
id({IDNR}) verbindt v(VV) met v(WN) en	zie R33
{(SCNR; GLNR)} verbindt v(AFD) met v(WN) en	zie R52
{(CPRS; IDNR)} verbindt v(PLR) met v(NMP) en	zie R43
{(DDHA; NAWH)} verbindt v(PLR) met v(HA) en	zie R45
id({NRHA, NAWH}) verbindt v(PAT) met v(HA) en	zie R09
id({NRHA}) verbindt v(VV) met v(HA) en	zie R34
id({ANR}) verbindt v(MW) met v(AFD) en	zie R11
id({ANR}) verbindt v(RU) met v(AFD) en	zie R57
id({BCD}) verbindt v(PB) met v(BEH) en	zie R83
id({MCD}) verbindt v(MVS) met v(MED) en	zie R91
id({RNR}) verbindt	
v(OPN) met $\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) \neq$ 'GEPL' en $t(\mathrm{SRTR}) =$ 'VPR' },	zie R65
v(OVR) met $\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) \neq$ 'GEPL' en $t(\mathrm{SRTR}) =$ 'VPR' } en	zie R74
v(PB) met $\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) \neq$ 'GEPL' en $t(\mathrm{SRTR}) =$ 'BHR' }, en	zie R84
id({WPL}) verbindt	
$\{t(\mathrm{NAW}) \mid t \in v(\mathrm{MW})$ en $t(\mathrm{SRTM}) \neq$ 'SP' } met v(PLR),	zie R24
$\{t(\mathrm{NAW}) \mid t \in v(\mathrm{PAT})\}$ met v(PLR) en	zie R08
$\{t(\mathrm{NAWH}) \mid t \in v(\mathrm{HA})\}$ met v(PLR), en	zie R37
id({IDNR}) verbindt	
v(SP) bilateraal met $\{t \in v(\mathrm{MW}) \mid t(\mathrm{SRTM}) =$ 'SP' },	zie R16
v(WN) bilateraal met $\{t \in v(\mathrm{MW}) \mid t(\mathrm{SRTM}) \neq$ 'SP' },	zie R19
v(MP) bilateraal met $\{t \in v(\mathrm{MW}) \mid t(\mathrm{SRTM}) =$ 'MP' } en	zie R28
v(NMP) bilateraal met $\{t \in v(\mathrm{MW}) \mid t(\mathrm{SRTM}) =$ 'NMP' }, en	zie R30
id({PNR, OPND}) verbindt	
v(ONT) bilateraal met $\{t \in v(\mathrm{OPN}) \mid t(\mathrm{ONTS}) =$ 'JA' } en	zie R67
$(\sum t \in v(\mathrm{SP}) : t(\mathrm{ABD})) \leq$	zie R13
(11/10) * ($\sum t \in v(\mathrm{RU})$ en $t(\mathrm{SRTR}) =$ 'VPR' : t(ABD)) en	,,
$(\forall t \in v(\mathrm{NMP}) \bowtie v(\mathrm{WN}) : t(\mathrm{VERG}) \leq t(\mathrm{SAL}))$ en	zie R31
$(\forall t \in v(\mathrm{OSP}) \cup v(\mathrm{OVR}) : \exists t' \in v(\mathrm{OPN}) :$	zie R70 en R73
($t'(\mathrm{PNR}) = t(\mathrm{PNR})$ en $t'(\mathrm{OPND}) < t(\mathrm{BDAT})$ en	,,
als $t'(\mathrm{ONTS}) =$ 'JA' dan $t(\mathrm{BDAT}) \leq t'(\mathrm{ONTD})))\}$.	,,

OPGAVEN

4.1. (a) Is "de lege toestand" toegestaan? (M.a.w. $(\lambda E \in \text{He}\,(\text{UZKH}): \varnothing) \in$ UZKH?)

(b) Zij UZKH2 het DB-universum gedefinieerd door in de definitie van UZKH het "≤"-teken in R13 te vervangen door het "<"-teken. Is "de lege toestand" in UZKH2 toegestaan?

4.2. Zij GZKH het DB-skelet waarover het DB-universum UZKH is gedefinieerd.

(a) Schrijf dom (GZKH) uit.

(b) Schrijf GZKH (MW) uit.

4.3. Ga voor elk van de volgende beweringen de geldigheid (en eventueel zelfs de zinledigheid) in UZKH na. Noem voor elke geldige bewering de constraint(s) waaruit de bewering volgt. Geef voor elke niet altijd geldige of zinledige bewering kort en duidelijk aan welke veranderingen in de definitie van het DB-universum nodig zijn om alsnog aan het gestelde te kunnen voldoen.

(a) Eenzelfde behandeling kan op eenzelfde datum op eenzelfde patient niet meer dan éénmaal plaatsvinden.

(b) Voor behandelsoort KNO geldt een vast tarief van 40 gulden.

(c) Voor elke behandelsoort geldt een eigen vast tarief.

(d) Afdeling 9 heeft minstens twintig medewerkers.

(e) Afdeling 1 heeft minstens twintig medewerksters.

(f) Afdeling 7 heeft minstens twintig werkneemsters.

(g) De afdeling neurologie heeft minstens vijf werkneemsters.

(h) Een specialist kan tot meer dan één afdeling behoren.

(i) Een specialist voor nul uren in de week is niet meer in dienst.

(j) Een specialist met toegezegde bedden is nog in dienst.

(k) Een patiënt kan voor meer dan één reden worden opgenomen.

(l) Alleen de "lopende" opnamen worden bijgehouden.

(m) Een patiënt is iemand die een opname, een behandeling of een medicijnverstrekking heeft ondergaan.

(n) Het tarief van een behandeling ligt vast door de combinatie van behandelcode en behandelend specialist.

(o) Het tarief van een behandeling ligt vast door de combinatie van behandelsoort en behandelend specialist.

(p) Er kan worden nagegaan of een specialist actief is geweest.
(q) Een medewerker kan tevens een patiënt zijn.
(r) Per medicijnverstrekking ligt de totaal verstrekte hoeveelheid vast.
(s) Elke behandeling vindt plaats tijdens een opname.
(t) Niemand is contactpersoon voor twee of meer plaatsen tegelijk.
(u) Een contactpersoon voor een plaats in de regio moet ook in die plaats wonen.
(v) Een specialist kan tevens groepsleider zijn.
(w) Niemand kan chef van meer dan één afdeling zijn.
(x) De hoofdbehandelaar en de assistent-specialist bij een patiëntbehandeling zijn altijd aan dezelfde afdeling verbonden.
(y) Een patiëntbehandeling vindt altijd plaats in de afdeling waaraan de hoofdbehandelaar is verbonden.
(z) Op elke verpleegruimte zijn er niet meer lopende opnamen dan er zich bedden in die verpleegruimte bevinden.

4.4. Is $\{v(\text{PAT}) \mid v \in \text{UZKH}\}$ in BCNF of in 3NF?
(U mag er hierbij van uitgaan dat $\text{Ms}(\{v(\text{PAT}) \mid v \in \text{UZKH}\}) = \{\{\text{PNR}\}\}$.)

4.5. Levert toevoeging van een willekeurig specialisttupel t aan "de lege toestand" weer een "correcte" DB-toestand (dat wil zeggen een element van UZKH) op?
Zo niet, schets dan de eisen aan de andere tupels die er minimaal moeten worden toegevoegd om een correcte DB-toestand te verkrijgen.

4.6. Wanneer we twee volgens UZKH "correcte" toestanden bij elkaar voegen, krijgen we dan weer een correcte toestand? Formeler verwoord luidt de vraag:
$\forall v \in \text{UZKH} : \forall v' \in \text{UZKH} : (\lambda E \in \text{He(UZKH)} : v(E) \cup v'(E)) \in \text{UZKH}$?

□

4.3 ENIGE DATABASE-FUNCTIES

In deze paragraaf definiëren we een 34-tal relevante DB-functies over UZKH. Op één uitzondering na is hierbij de naam van een DB-functie met betrekking tot een geordend paar tabelindexen $(\alpha ; \beta)$ van de vorm "$\text{H}\alpha\beta$".

Hspsp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{SP}) \times v(\mathrm{SP}) \mid t(\mathrm{LOC}) = u(\mathrm{IDNR})\}$
Hwnwn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{WN}) \times v(\mathrm{WN}) \mid t(\mathrm{GLNR}) = u(\mathrm{IDNR})\}$
Hopnpat $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OPN}) \times v(\mathrm{PAT}) \mid t(\mathrm{PNR}) = u(\mathrm{PNR})\}$
Hpbpat $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PB}) \times v(\mathrm{PAT}) \mid t(\mathrm{PNR}) = u(\mathrm{PNR})\}$
Hmvspat $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{MVS}) \times v(\mathrm{PAT}) \mid t(\mathrm{PNR}) = u(\mathrm{PNR})\}$
Hafdmw $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{AFD}) \times v(\mathrm{MW}) \mid t(\mathrm{CHNR}) = u(\mathrm{IDNR})\}$
Hpbsp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PB}) \times v(\mathrm{SP}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hpbspa $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PB}) \times v(\mathrm{SP}) \mid t(\mathrm{ASNR}) = u(\mathrm{IDNR})\}$
Hopnsp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OPN}) \times v(\mathrm{SP}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hospsp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OSP}) \times v(\mathrm{SP}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hmvssp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{MVS}) \times v(\mathrm{SP}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hvvwn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{VV}) \times v(\mathrm{WN}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hplrnmp $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PLR}) \times v(\mathrm{NMP}) \mid t(\mathrm{CPRS}) = u(\mathrm{IDNR})\}$
Hplrha $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PLR}) \times v(\mathrm{HA}) \mid t(\mathrm{DDHA}) = u(\mathrm{NAWH})\}$
Hpatha $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PAT}) \times v(\mathrm{HA}) \mid t(\mathrm{NRHA}) = u(\mathrm{NRHA})\}$
Hvvha $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{VV}) \times v(\mathrm{HA}) \mid t(\mathrm{NRHA}) = u(\mathrm{NRHA})\}$
Hmwafd $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{MW}) \times v(\mathrm{AFD}) \mid t(\mathrm{ANR}) = u(\mathrm{ANR})\}$
Hruafd $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{RU}) \times v(\mathrm{AFD}) \mid t(\mathrm{ANR}) = u(\mathrm{ANR})\}$
Hpbbeh $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PB}) \times v(\mathrm{BEH}) \mid t(\mathrm{BCD}) = u(\mathrm{BCD})\}$
Hmvsmed $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{MVS}) \times v(\mathrm{MED}) \mid t(\mathrm{MCD}) = u(\mathrm{MCD})\}$
Hopnru $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OPN}) \times v(\mathrm{RU}) \mid t(\mathrm{RNR}) = u(\mathrm{RNR})\}$
Hovrru $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OVR}) \times v(\mathrm{RU}) \mid t(\mathrm{RNR}) = u(\mathrm{RNR})\}$
Hpbru $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PB}) \times v(\mathrm{RU}) \mid t(\mathrm{RNR}) = u(\mathrm{RNR})\}$
Hspmw $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{SP}) \times v(\mathrm{MW}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hwnmw $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{WN}) \times v(\mathrm{MW}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$

Hmpwn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{MP}) \times v(\mathrm{WN}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hnmpwn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{NMP}) \times v(\mathrm{WN}) \mid t(\mathrm{IDNR}) = u(\mathrm{IDNR})\}$
Hafdwn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{AFD}) \times v(\mathrm{WN}) \mid t(\mathrm{SCNR}) = u(\mathrm{IDNR})\}$

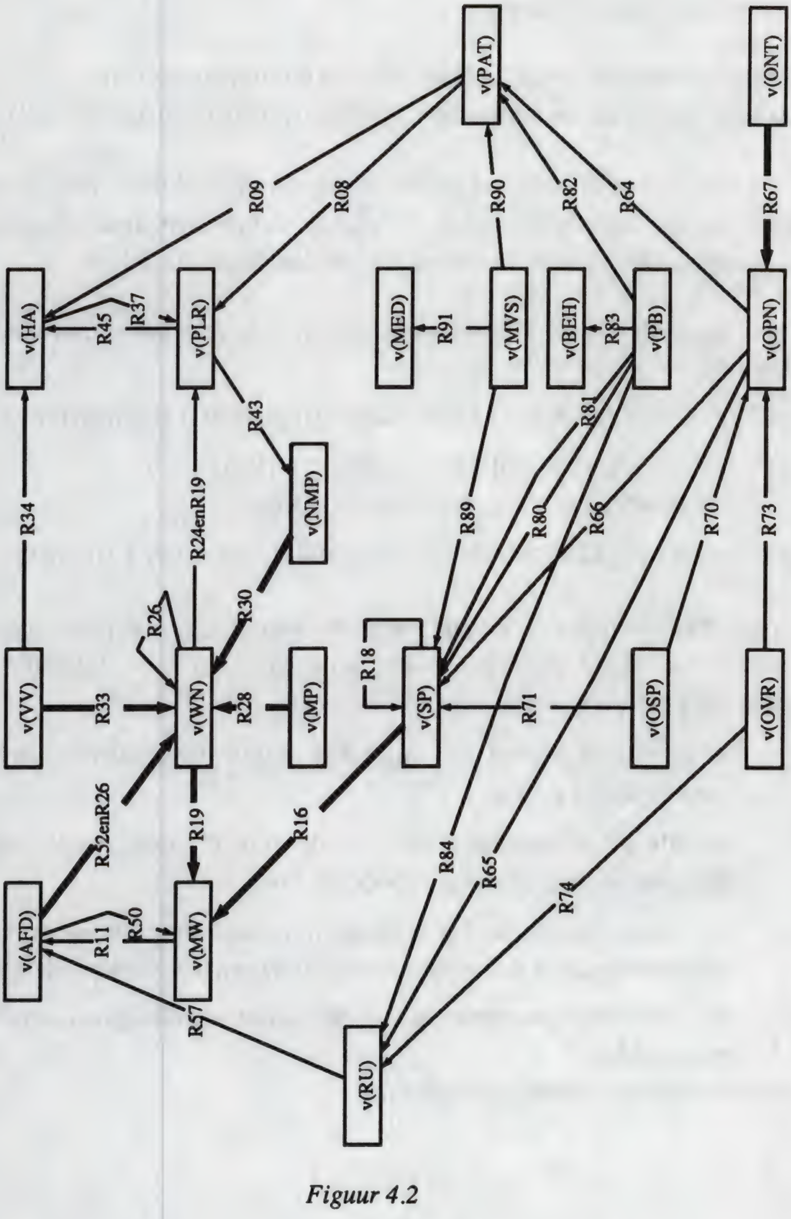
Hontopn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{ONT}) \times v(\mathrm{OPN}) \mid t \restriction \{\mathrm{PNR}, \mathrm{OPND}\} = u \restriction \{\mathrm{PNR}, \mathrm{OPND}\}\}$

Hpatplr $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{PAT}) \times v(\mathrm{PLR}) \mid t(\mathrm{NAW})\,(\mathrm{WPL}) = u(\mathrm{WPL})\}$
Hhaplr $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{HA}) \times v(\mathrm{PLR}) \mid t(\mathrm{NAWH})\,(\mathrm{WPL}) = u(\mathrm{WPL})\}$

Hwnplr $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{WN}) \times v(\mathrm{PLR}) \mid \exists m \in v(\mathrm{MW})\colon m(\mathrm{IDNR}) = t(\mathrm{IDNR})$ en
$m(\mathrm{NAW})\,(\mathrm{WPL}) = u(\mathrm{WPL})\}$

Hospopn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OSP}) \times v(\mathrm{OPN}) \mid t(\mathrm{PNR}) = u(\mathrm{PNR})$ en $u(\mathrm{OPND}) < t(\mathrm{BDAT})$ en
als $u(\mathrm{ONTS}) =$ 'JA' dan $t(\mathrm{BDAT}) \leq u(\mathrm{ONTD}$
Hovropn $= \lambda v \in \mathrm{UZKH}\colon \{(t\,;u) \in v(\mathrm{OVR}) \times v(\mathrm{OPN}) \mid t(\mathrm{PNR}) = u(\mathrm{PNR})$ en $u(\mathrm{OPND}) < t(\mathrm{BDAT})$ en
als $u(\mathrm{ONTS}) =$ 'JA' dan $t(\mathrm{BDAT}) \leq u(\mathrm{ONTD}$

De onderstaande figuur geeft voor elk van de zojuist gedefinieerde DB-functies H aan hoe de functie $H(v)$ in toestand v "loopt". Gezien de systematische naamgeving voor deze DB-functies bevat deze figuur (in tegenstelling tot Figuur 2.11) niet de namen van de betreffende DB-functies maar de belangrijkste constraints voor het bestaan van die functies. De in Opgave 4.9 genoemde injectieve functies zijn gerepresenteerd door een "vette" pijl.



Figuur 4.2

OPGAVEN

4.7. Bewijs de zojuist gedefinieerde functies inderdaad DB-functies over UZKH zijn. (We merken hierbij op dat we deze 34 functies door middel van lege regels in zes groepen hebben verdeeld, en wel op grond van de nu te geven bewijsvoeringen.)

4.8. Zij Hpatha2 = $\lambda v \in$ UZKH: $\{(t\,;\,u) \in v(\mathrm{PAT}) \times v(\mathrm{HA}) \mid t(\mathrm{NAWH}) = u(\mathrm{NAWH})\}$.
Bewijs dat Hpatha2 = Hpatha.

4.9. Bewijs dat voor elke $v \in$ UZKH de volgende functies injectief zijn:
Hafdwn(v), Hmpwn(v), Hnmpwn(v), Hontopn(v), Hspmw(v), Hvvwn(v), Hwnmw(v).

4.10. Geef een formele definitie van de DB-functie over UZKH die in elke toestand aan elk tupel van een verplicht verzekerde werknemer het tupel van diens woonplaats toevoegt en bewijs dat deze functie inderdaad een DB-functie over UZKH is.

4.11. Geef, uitgaande van een DB-toestand $v \in$ UZKH, de volgende verzamelingen in woorden weer:

(a) $(((v(\mathrm{PB}) \restriction \{\mathrm{PNR}, \mathrm{BDAT}\}) \bowtie v(\mathrm{OVR})) \restriction \{\mathrm{PNR}\}) \bowtie \mathrm{rng}(\mathrm{Hmvspat}(v))$.

(b) $\{t \mid t \in v(\mathrm{WN}) \text{ en } t(\mathrm{GLNR}) = t(\mathrm{IDNR})\} \restriction \{\mathrm{IDNR}\}$
$- (v(\mathrm{WN}) \bowtie v(\mathrm{VV})) \infty \{(\mathrm{IDNR};\, \mathrm{GLNR})\}$.

(c) $((v(\mathrm{PB}) \infty \{(\mathrm{LOC};\, \mathrm{ASNR}), (\mathrm{IDNR};\, \mathrm{IDNR})\}) \bowtie v(\mathrm{SP}) \restriction \{\mathrm{IDNR}\}$.

4.12. Onder BMD(v,t), *het (beknopte) medische dossier* van een patiënttupel t in DB-toestand $v \in$ UZKH, verstaan we een functie over {ALGGEG, MVSGEG, PBHGEG, OPNGEG} met als waarden (achtereenvolgens)

- het betreffende patiënttupel "uitgebreid" met de (twee) telefoonnummers van de huisarts van die patiënt,
- van alle medicijnverstrekkingen aan die patiënt: medicijncode, ingangsdatum, frequentie per dag en aantal eenheden per keer,
- van alle behandelingen van die patiënt: behandelcode en -naam, volgnummer, behandelingsdatum en het identiteitsnummer van de behandelende specialist,
- van alle beëindigde opnamen van die patiënt: opnamedatum, ontslagdatum en opnamereden.

Geef een formele definitie van BMD(v,t).

□

5 DYNAMISCHE CONSTRAINTS

5.1 TRANSITIERELATIES

Met de keuze van een DB-universum U willen we vastleggen welke *toestanden* in een organisatie formeel zijn toegestaan. In het algemeen echter zijn niet alle toestands*overgangen* (alias *transities*) tussen die op zich toegestane toestanden geoorloofd in die organisatie. We kunnen nu de verzameling toegestane transities vastleggen door middel van een deelverzameling R van $U \times U$ met als intuïtieve betekenis:

$(v\,;v') \in R \Longleftrightarrow$ de directe overgang van toestand v naar toestand v' is toegestaan.

Als $(v\,;v') \notin R$ dan kan het nog wel zo zijn dat v' *indirect* vanuit v te bereiken is, namelijk als $(v\,;v') \in \mathrm{Tcl}(R)$, dat willen zeggen als v' via een aantal toegestane tussenstappen te bereiken is vanuit v (zie Definitie 0.22).

Een element van $U \times U$ noemen we een *transitie binnen U* en een verzameling toegestane transities behorende bij het DB-universum U noemen we wel een *transitierelatie op U*. Meer in het algemeen:

DEFINITIE 5.1:
Als U een verzameling is, dan:

(a) p is een **transitie binnen** U $\overset{D}{\Longleftrightarrow}$ $p \in U \times U$;

(b) R is een **transitierelatie op** U $\overset{D}{\Longleftrightarrow}$ $R \subseteq U \times U$.

Meestal is het gewenst dat voor elke toestand $v \in U$ de transitie van v naar zichzelf ook geoorloofd is. Zo'n "pas op de plaats" kan bijvoorbeeld optreden bij een poging tot

verwijdering van een tupel dat niet in v blijkt voor te komen. Kortom, meestal willen we dat zo'n transitierelatie R *reflexief op* U is, dat wil zeggen $\forall v \in U : (v\,; v) \in R$.

VOORBEELD 5.1:

Stel dat er in de organisatie met het hieronder beschreven DB-universum VBU, afkomstig uit Voorbeeld 1.3, de volgende eisen moeten gelden:

(DC1) Er mogen geen bestaande afdelingsnummers vervallen (hoewel bijvoorbeeld de naam van een afdeling wel mag veranderen).

(DC2) Een medewerker moet altijd aan dezelfde afdeling verbonden blijven.

(DC3) Salarissen van medewerkers mogen niet afnemen.

Om de constraints (DC2) en (DC3) te kunnen formaliseren, zullen we aannemen dat een medewerker "in de loop der tijd" hetzelfde identiteitsnummer behoudt. We kunnen nu deze constraints formeel weergeven door middel van de hieronder gedefinieerde transitierelatie VBR op VBU. (Volledigheidshalve herhalen we ook nog even de definitie van het DB-universum VBU.)

FM = {(NR ; $\mathbb{N}$),
(NAAM ; Chs (40)),
(SAL ; $\mathbb{N}$),
(GESL ; {'M' , 'V'}),
(AFDNR; [1 .. 99])};

FA = {(ANR ; $\mathbb{N}$),
(NAAM ; Chs (45)),
(MANNR ; $\mathbb{N}$)};

WM = {T | $T \subseteq \Pi$(FM) en {NR} is u.i. in T};

WA = {T | $T \subseteq \Pi$(FA) en {ANR} is u.i. in T en {NAAM} is u.i. in T};

HF = {(MEDEW ; WM),
(AFD ; WA)};

VBU = {v | $v \in \Pi$(HF) en
{(AFDNR; ANR)} verbindt v(MEDEW) met v(AFD) en
{(MANNR; NR)} verbindt v(AFD) met v(MEDEW)};

$$
\begin{aligned}
\text{VBR} = \{(v\,;v') \mid (v\,;v') \in \text{VBU} \times \text{VBU en}& \\
v(\text{AFD}) \restriction \{\text{ANR}\} \subseteq v'(\text{AFD}) \restriction \{\text{ANR}\} \text{ en}& \qquad \text{(DC1)} \\
\forall t \in v(\text{MEDEW}): \forall t' \in v'(\text{MEDEW}):& \\
[\text{als } t(\text{NR}) = t'(\text{NR}) \text{ dan } (t(\text{AFDNR}) = t'(\text{AFDNR})& \qquad \text{(DC2)} \\
\text{en } t(\text{SAL}) \leq t'(\text{SAL}))]\}& \qquad \text{(DC3)}
\end{aligned}
$$

□ Voorbeeld 5.1.

De eisen waaraan de *toestanden* in een organisatie moeten voldoen, noemen we *statische constraints* en de eisen waaraan de *transities* moeten voldoen, noemen we wel *dynamische constraints*.

We willen er op wijzen dat het soms onwenselijk is om dynamische constraints als "keiharde" constraints op te vatten. Het kan namelijk voorkomen dat een bepaalde transitie achteraf onjuist of onbedoeld bleek te zijn maar helaas wegens de dynamische constraints niet meer ongedaan kan worden gemaakt. Als bijvoorbeeld reeds aanwezige afdelingsnummers (of ordernummers) niet meer mogen vervallen, dan kan het abusievelijk toevoegen van een afdeling (of een order) niet meer ongedaan worden gemaakt. We zeggen in dat geval wel dat de transitie *irreversibel* is:

DEFINITIE 5.2:
Als $(x\,;y)$ een geordend paar is en R is een verzameling geordende paren, dan:
(a) $(x\,;y)$ **is reversibel in** R $\overset{D}{\Longleftrightarrow}$ $(y\,;x) \in \text{Tcl}(R)$;
(b) $(x\,;y)$ **is irreversibel in** R $\overset{D}{\Longleftrightarrow}$ $(y\,;x) \notin \text{Tcl}(R)$.

Als R uitsluitend reversibele elementen bevat, dan noemen we R zelf ook reversibel:

DEFINITIE 5.3:
Als R een verzameling geordende paren is, dan:
R **is reversibel** $\overset{D}{\Longleftrightarrow}$ $\forall p \in R : p$ is reversibel in R.

Bij niet-reversibele transitierelaties wordt een dynamische constraint vaak niet gebruikt als "keiharde" constraint maar meer als waarschuwingscriterium ("Wilt u dit écht?") of als een voorwaarde waaronder er speciale toestemming nodig is om de wijziging toch door te voeren. Dit betekent dus dat deze dynamische constraints dan niet als echte transitie-invarianten gebruikt kunnen worden.

In het volgende voorbeeld bespreken we enige algemene vormen van dynamische constraints. Elke algemene vorm wordt tevens toegelicht aan de hand van VBU of VBR uit Voorbeeld 5.1.

VOORBEELD 5.2:

We gaan uit van een DB-skelet g, een DB-universum U over g, $E \in \text{dom}(g)$, $S \subseteq g(E)$, $B \subseteq g(E)$, $C \subseteq g(E)$ en $a \in g(E)$, waarbij de a-waarden voor E getallen zijn. We zullen de constraint-vormen uitdrukken in termen van een willekeurige transitie $(v\,;\,v')$ binnen U.

(V1) De E-tabel is *cumulatief*, dat wil zeggen alle E-gegevens blijven ongewijzigd aanwezig. Kortom, er mogen geen E-tupels vervallen:

$v(E) \subseteq v'(E)$.

Het is interessant op te merken dat deze dynamische constraint is op te vatten als een verbindingseis "door de tijd heen":

id $(g(\mathrm{E}))$ verbindt $v(E)$ met $v'(E)$.

Voor E = AFD in VBU zou deze eis nogal wat inhouden, namelijk dat een afdeling(snummer) niet mag vervallen, dat de bijbehorende afdelingsnaam niet mag veranderen en dat ook de manager van die afdeling niet mag worden vervangen. In feite is dus het toevoegen van geheel nieuwe afdelingen de enig toegestane verandering op de afdelingentabel.
Wanneer de E-tabel voor bijvoorbeeld logging-activiteiten bedoeld is, dan kan deze zware eis van cumulativiteit wel realistisch zijn.

(V2) Er mag geen enkele combinatie van waarden van *primaire* attributen - dat wil zeggen attributen die element van een minimale sleutel zijn (zie de definities 2.13 en 2.19) - in de E-tabel vervallen:

id(Prim $(E\,,U)$) verbindt $v(E)$ met $v'(E)$.

Voor E = AFD in VBU betekent dit:

{ANR, NAAM} verbindt v(AFD) met v'(AFD).

Met andere woorden, een afdelingsnummer mag niet vervallen en de

bijbehorende afdelingsnaam mag niet veranderen; de manager mag echter wel worden vervangen (en ook mogen er nieuwe afdelingen bijkomen).

Merk op dat de eis (V1) de eis (V2) impliceert.

(V3) De E-tabel is *cumulatief met betrekking tot S*, dat wil zeggen er mogen geen S-waarden vervallen:

$$v(E) \upharpoonright S \subseteq v'(E) \upharpoonright S.$$

We kunnen dit ook weer formuleren als een verbindingseis:

id(S) verbindt $v(E)$ met $v'(E)$.

Een voorbeeld van deze vorm van dynamische constraint is de eis (DC1) uit Voorbeeld 5.1.

Als S uitsluitend primaire attributen bevat, bijvoorbeeld als S een minimale sleutel van E in U is, dan geldt $S \subseteq \mathrm{Prim}(E, U)$; in dat geval impliceert de eis (V2) de eis (V3).

(V4) Bij elke B-waarde die aanwezig blijft in de E-tabel moeten de bijbehorende C-waarden constant blijven:

$$\forall t \in v(E) : \forall t' \in v'(E)\text{: als } t \upharpoonright B = t' \upharpoonright B \text{ dan } t \upharpoonright C = t' \upharpoonright C.$$

De eis (DC2) uit Voorbeeld 5.1 is een speciaal geval van deze vorm. (Neem U = VBU, E = MEDEW, B = {NR} en C = {AFDNR}.)

Als B een sleutel van E in U is en $S = B \cup C$ in (V3), dan impliceert die eis (V3) de eis (V4).

(V5) De a-waarde van elk E-tupel in de nieuwe toestand moet ten minste even groot zijn als de a-waarde van elk E-tupel in de oude toestand met dezelfde S-waarde:

$$\forall t \in v(E) : \forall t' \in v'(E)\text{: als } t \upharpoonright S = t' \upharpoonright S \text{ dan } t(a) \leq t'(a).$$

Bij deze vorm zal S meestal een sleutel van E in U zijn, zoals bijvoorbeeld het geval is bij de eis (DC3) uit Voorbeeld 5.1.

Als $B = S$ en $C = \{a\}$ in (V4), dan impliceert die eis (V4) de eis (V5).

(V6) Het aantal E-tupels mag niet afnemen:

$$\mid v(E) \mid \leq \mid v'(E) \mid .$$

Voor E = AFD in VBU zou dit betekenen dat afdelingen niet zonder meer mogen vervallen maar kennelijk wel mogen worden vervangen door andere afdelingen.

Als S in (V3) een sleutel van E in U is, dan impliceert de eis (V3) de eis (V6).

□ Voorbeeld 5.2.

In de voorbeelden 5.1 en 5.2 zagen we dat er soms voor één van de minimale sleutels van E in U een speciale rol is weggelegd. De achterliggende gedachte is dat de waarden voor die speciale sleutel in de loop der tijd één-één-duidig blijven corresponderen met de gerepresenteerde "real-world" objecten zelf. We noemen zo'n speciale minimale sleutel wel een *primary key*. De andere minimale sleutels noemen we in dit verband wel *alternate keys*. Hoewel het onderscheid tussen primary keys en alternate keys in de literatuur meestal op andere (en vaak niet altijd duidelijke) gronden wordt gemaakt, lijkt ons de zojuist genoemde reden nog het meest zinvolle argument voor zo'n onderscheid. Een nadere discussie over dit onderwerp kan de lezer onder meer vinden in Hoofdstuk 3 van [Da 86b].

In Voorbeeld 5.2 hebben we tevens gezien dat het verbindingsbegrip niet alleen van dienst is voor de formulering van statische constraints maar ook voor die van dynamische constraints. In die gevallen is de verbindende attributentransformatie uiteraard een identieke functie.

Voor de klasse van dynamische constraints die we in (V4) van Voorbeeld 5.2 tegenkwamen, willen we een aparte naamgeving en notatie invoeren. We zullen daarom in onderstaand geval *C transitie-afhankelijk van B bij* $(T\,;T')$ noemen, naar analogie van momentane afhankelijkheid.

DEFINITIE 5.4:
Als A, B en C verzamelingen zijn en T en T' zijn tabellen over A, dan:
$B \overset{\sim}{\rightarrow} C$ **bij** $(T\,;T') \overset{D}{\Longleftrightarrow} \forall t \in T : \forall t' \in T'$: als $t \upharpoonright B = t' \upharpoonright B$ dan $t \upharpoonright C = t' \upharpoonright C$.

De eis (V4) van Voorbeeld 5.2 kunnen we nu dus schrijven als: $B \overset{\sim}{\rightarrow} C$ bij $(v(E)\,; v'(E))$.

Het volgende lemma is het analogon van Lemma 2.7 en geeft enige elementaire eigenschappen van transitie-afhankelijkheid weer.

LEMMA 5.1:
Als A een verzameling is en T en T' zijn tabellen over A en $B \subseteq A$ en $C \subseteq A$ en $D \subseteq A$, dan:

(a) als $C \subseteq B$, dan $B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T')$;

(b) als $B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T')$ en $C \stackrel{\sim}{\rightarrow} D$ bij $(T\,;T')$, dan $B \stackrel{\sim}{\rightarrow} D$ bij $(T\,;T')$;

(c) $B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T')$ desda $\forall c \in C : B \stackrel{\sim}{\rightarrow} \{c\}$ bij $(T\,;T')$.

Lemma 5.2 relateert transitie-afhankelijkheid aan momentane afhankelijkheid (en in (c) aan zichzelf).

LEMMA 5.2:
Als A een verzameling is en T en T' zijn tabellen over A en $B \subseteq A$ en $C \subseteq A$, dan:

(a) $B \rightarrow C$ in $T \iff B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T)$;

(b) $B \rightarrow C$ in $T \cup T' \iff B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T')$ en $B \rightarrow C$ in T en $B \rightarrow C$ in T';

(c) $B \stackrel{\sim}{\rightarrow} C$ bij $(T\,;T') \iff B \stackrel{\sim}{\rightarrow} C$ bij $(T'\,;T)$.

De bewijzen van deze lemma's worden als opgaven aan de lezer overgelaten. Ter illustratie van onderdeel (b) van Lemma 5.2 merken we op dat in Voorbeeld 5.1 de dynamische constraint DC2 samen met de uniciteitseis in de definitie van WM equivalent is met de *dynamische* constraint dat $\{NR\} \rightarrow \{AFDNR\}$ in $v(MEDEW) \cup v'(MEDEW)$.

We willen het begrip transitie-afhankelijkheid vooral ook definiëren op het niveau van transitierelaties: We zullen in onderstaand geval *C constant afhankelijk van B in E bij R* noemen.

DEFINITIE 5.5:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{dom}(g)$ en B en C zijn verzamelingen en $R \subseteq U \times U$, dan:
$B \stackrel{\sim}{\rightarrow} C$ **in** E **bij** $R \stackrel{D}{\iff} \forall (v\,;v') \in R : B \stackrel{\sim}{\rightarrow} C$ bij $(v(E)\,;v'(E))$.

Een eenvoudige maar belangrijke constatering is dat bij reflexieve transitierelaties permanente afhankelijkheid uit constante afhankelijkheid volgt:

STELLING 5.1:
Als g een verzamelingsfunctie is en U is een DB-universum over g en $E \in \mathrm{dom}(g)$ en $B \subseteq g(E)$ en $C \subseteq g(E)$ en $R \subseteq U \times U$, dan:
als $B \overset{\sim}{\rightarrow} C$ in E bij R en R is reflexief op U dan $B \overset{\wedge}{\rightarrow} C$ in E van U.

Bewijs:
Te bewijzen dat $\forall v \in U : B \rightarrow C$ in $v(E)$;

$$
\begin{aligned}
& B \overset{\sim}{\rightarrow} C \text{ in } E \text{ bij } R \\
\Leftrightarrow\ & \forall (v\,;v') \in R : B \overset{\sim}{\rightarrow} C \text{ bij } (v(E)\,; v'(E)) && \text{(volgens Definitie 5.5)} \\
\Rightarrow\ & \forall v \in U : B \overset{\sim}{\rightarrow} C \text{ bij } (v(E)\,; v(E)) && \text{(omdat } R \text{ reflexief op } U \text{ is)} \\
\Leftrightarrow\ & \forall v \in U : B \rightarrow C \text{ in } v(E) && \text{(volgens Lemma 5.2 (a))}
\end{aligned}
$$

□

OPGAVEN

5.1. Is VBR reflexief op VBU in Voorbeeld 5.1?

5.2. In Voorbeeld 5.1 denken we aan FM het geordend paar (ST; {'O', 'H', 'S', 'W'}) toegevoegd; hierbij staat ST voor status, 'O' ongehuwd, 'H' voor gehuwd, 'S' voor gescheiden en 'W' voor weduwe of weduwnaar. Geef formeel weer welke eisen er aan de definitie van VBR zouden moeten worden toegevoegd.

5.3. Geef de volgende dynamische constraints voor een transitie $(v\,; v')$ binnen VBU uit Voorbeeld 5.1 formeel weer.

(a) Het aantal medewerkers van een afdeling mag niet met meer dan vijf personen in één keer stijgen.

(b) Het aantal medewerkers van een bestaande afdeling mag niet met meer dan vijf personen in één keer stijgen.

(c) Het aantal medewerkers van een manager mag niet met meer dan vijf personen in één keer stijgen.

(d) Het aantal medewerkers van een "prille" manager mag niet meer dan vijf personen bedragen.

5.4. (a) Leg uit waarom VBR uit Voorbeeld 5.1 niet reversibel is.

(b) Geef (desondanks) een reversibel element $(v\,;\,v')$ van VBR zodanig dat $v \neq v'$.

(c) Leidt het verhogen van een salaris van een medewerker tot een irreversibele transitie in VBR?

5.5. In Voorbeeld 5.2 eindigen (V2) tot en met (V6) elk met een implicatie. Bewijs die vijf implicaties.

5.6. Geef voor elk van de constraint-vormen (V1) tot en met (V6) in Voorbeeld 5.2 aan onder welke (eventuele) extra voorwaarden een transitie $(v\,;\,v)$ met $v \in$ VBU aan die constraint-vorm voldoet.

5.7. Vervang in de verwoording van de eisen in (V1), (V2) en (V3) van Voorbeeld 5.2 het woord "vervallen" door het woord "bijkomen" en geef deze drie nieuwe eisen weer met behulp van het verbindingsbegrip.

5.8. Is de eis (V2) van Voorbeeld 5.2 equivalent met de uitspraak dat er geen enkele waarde van een minimale sleutel mag vervallen?
Geef een bewijs of een tegenvoorbeeld voor elk van de twee gevraagde implicaties.

5.9. Bewijs Lemma 5.1.

5.10. Formuleer en bewijs het analogon van Lemma 2.8.

5.11. Als A een verzameling is en T en T' zijn tabellen over A en $B \subseteq A$ en $C \subseteq A$, dan:

(a) $B \tilde{\rightarrow} C$ bij $(\varnothing\,;\,T')$;

(b) $B \tilde{\rightarrow} \varnothing$ bij $(T\,;\,T')$;

(c) $\varnothing \tilde{\rightarrow} C$ bij $(T\,;\,T')$ desda $T = \varnothing$ of $T' = \varnothing$ of $|\, T \upharpoonright C \cup T' \upharpoonright C \,| \leq 1$.

Bewijs dit analogon van Lemma 2.9.

5.12. Bewijs Lemma 5.2. Laat eveneens zien dat uit $B \tilde{\rightarrow} C$ bij $(T\,;\,T')$ nog niet volgt dat $B \rightarrow C$ in $T \cup T'$.

5.13. Als A een verzameling is en T en T' zijn tabellen over A en $B \subseteq A$ en $C \subseteq A$, dan:

(a) als $B \tilde{\rightarrow} C$ bij $(T\,;T')$, dan $B \rightarrow C$ in $(T \bowtie T' \upharpoonright B) \cup (T' \bowtie T \upharpoonright B)$;

(b) als B u.i. in T is en B is u.i. in T', dan:
$(B \tilde{\rightarrow} C$ bij $(T\,;T')$ en $T \upharpoonright B \subseteq T' \upharpoonright B) \Leftrightarrow T \upharpoonright (B \cup C) \subseteq T' \upharpoonright (B \cup C)$.

Bewijs dit.

5.14. We noemen een verzameling R van geordende paren *symmetrisch* desda $\forall (x\,;y) \in R : (y\,;x) \in R$.
Bewijs nu het volgende.
Als R een verzameling geordende paren is, dan:

(a) Als R symmetrisch is, dan is R reversibel.

(b) De volgende drie beweringen zijn equivalent:
(b1) R is reversibel;
(b2) Tcl(R) is reversibel;
(b3) Tcl(R) is symmetrisch.

(c) Als R reversibel is, dan $\forall p \in$ Tcl(R): p is reversibel in R.

□

5.2 EEN NIET-TRIVIAAL VOORBEELD VAN EEN TRANSITIERELATIE

Om een indruk te geven van de mogelijke (subtiele) verschijningsvormen van dynamische constraints in de context van een op zich reeds complex database-universum, zullen we in deze paragraaf een niet-triviale transitierelatie op het database-universum UZKH uit Hoofdstuk 4 definiëren.

We beginnen met een informele beschrijving van de dynamische constraints, vervolgen met een "semi-formele" tussenvorm van sommige van de meer subtiele constraints (als analyse-hulpmiddel) en eindigen met de formele definitie van de transitierelatie RZKH op UZKH. Evenals in Hoofdstuk 4 geldt ook hier (en meer in het algemeen) dat een informele beschrijving meestal niet voldoende duidelijk is om daar de formele definitie eenduidig uit te kunnen afleiden.

We beginnen met een informele beschrijving van de dynamische constraints voor het medische gedeelte en de daarmee verbonden administratie, direct gevolgd door die voor de personele aangelegenheden.

(C00) Een afdeling(snummer) mag niet vervallen; wel mag bijvoorbeeld de naam of de chef of de souschef van een afdeling veranderen. (Souschefs verwelken, chefs die vergaan, maar onze afdelingen blijven altijd bestaan.)

(C01) Alle medicijnverstrekkingsgegevens moeten ongewijzigd aanwezig blijven.

(C02) Ook alle patiëntbehandelingsgegevens moeten ongewijzigd aanwezig blijven.

(C03) Na invoering moeten van elke opname de volgende gegevens ongewijzigd aanwezig blijven: patiëntnummer, opnamedatum, ruimtenummer, identiteitsnummer van de in eerste instantie verantwoordelijke specialist en de gegevens van de (toenmalige) huisarts van de patiënt.

(C04) Van elke beëindigde opname moeten ook de andere opnamegegevens ongewijzigd aanwezig blijven, evenals de bijbehorende ontslaggegevens.

(C05) Bijna alle overdrachtgegevens moeten ongewijzigd aanwezig blijven: alleen de opnamereden kan nog worden gewijzigd.

(C06) Ook bijna alle overplaatsingsgegevens moeten ongewijzigd aanwezig blijven: alleen aan de verpleegkundige bevindingen kunnen nog opmerkingen worden *toegevoegd.* (Reeds aanwezige aantekeningen mogen namelijk niet meer worden gewijzigd of verwijderd.) Met het oog op de te geven formalisatie van deze eis wijzen we nog eens op de voorlaatste alinea vóór Definitie 0.21.

(C07) Een medicijn(code) blijft altijd van dezelfde soort.

(C08) Een behandeling blijft ook altijd van dezelfde soort.

(C09) Van elke patiënt die in de administratie aanwezig blijft, mogen de volgende gegevens niet meer worden gewijzigd: geboorteplaats, geboortedatum, inschrijfdatum, bloedgroep en rhesusfactor.

(C10) Voor mannelijke patiënten moet ook de naam ongewijzigd blijven.

(C11) Van een werknemer mag het salaris per halve dag per week niet afnemen.

(C12) Van een werknemer kan het aantal ziektedagen niet afnemen, met uitzondering van de momenten waarop voor *alle* werknemers het aantal ziektedagen weer op nul wordt gezet.

(C13) De vakbonden hebben kunnen afdwingen dat er niet mag worden bezuinigd op het aantal vrije dagen per jaar (op basis van een volledige betrekking) van een niet-medisch personeelslid dat verplicht verzekerd is.

(C14) Bij een wijziging in de gegevens over het patiëntenaantal, de vorige patiëntentoename of de verwachte patiëntentoename van een huisarts moet de nieuwe "patiëntentoename vorig jaar" gelijk zijn aan het verschil tussen het nieuwe aantal patiënten en het oude aantal patiënten van die huisarts. (In tegenstelling tot de eveneens als jaarlijks bedoelde wijzigingen genoemd in C12, die voor alle werknemers tegelijk moeten gebeuren, moeten de in C14 bedoelde wijzigingen afzonderlijk per huisarts kunnen geschieden.)

(C15) In verband met de continuïteit van een afdeling mag de leiding van een afdeling, dat wil zeggen chef plus souschef van die afdeling, niet in één keer worden vervangen door twee geheel nieuwe personen. (Wel mag bijvoorbeeld de chef verdwijnen en gelijktijdig de souschef de nieuwe chef van die afdeling worden.)

Wat betreft de "life-cycle" van een ruimte in ons ziekenhuis gelden de volgende regels en nevenvoorwaarden.

(C16) Alle "vervallen" ruimten in de nieuwe toestand moeten reeds in de oude toestand bekend zijn, maar dan niet met de status "gepland". (Een ruimte mag namelijk niet "vanuit het niets" meteen vervallen en van een geplande ruimte die er uiteindelijk niet komt worden de gegevens gewoon verwijderd.)

(C17) Elk ruimtenummer met de status "aanwezig" moet blijven bestaan, moet daarbij dezelfde oppervlakte houden en kan niet meer de status "gepland" krijgen.

(C18) Voor aanwezige behandelruimten en verpleegruimten moet bovendien de soort ongewijzigd blijven. (Recepties, magazijnen en "diverse" ruimten mogen wél van soort veranderen; ze kunnen in principe zelfs verpleegruimten of behandelruimten worden. Echter, verpleegruimten en behandelruimten die voor andere doeleinden gebruikt gaan worden, zullen administratief als "vervallen" worden beschouwd en onder een ander ruimtenummer verder gaan in ons ziekenhuis. We merken hierbij overigens op dat dergelijke administratieve constructies in de praktijk niet ongebruikelijk zijn.)

(C19) De gegevens van alle "vervallen" ruimten moeten in de nieuwe toestand ongewijzigd blijven bestaan.

De volgens C16 tot en met C19 voor een ruimtenummer enig toegestane statusovergangen hebben we schematisch weergegeven in het volgende diagram (waarbij "~" de situatie aangeeft dat het ruimtenummer niet voorkomt in de ruimte-tabel):

De mogelijke statusovergangen voor een ruimte(nummer)

De wel en niet toegestane statusovergangen kunnen we ook weergeven met behulp van een matrix:

nieuw / oud	~	GEPL	AANW	VERV
~	+	+	+	–
GEPL	+	+	+	–
AANW	–	–	+	+
VERV	–	–	–	+

De mogelijke statusovergangen in matrixvorm

De reden voor een niet toegestane statusovergang en de eventuele nevenvoorwaarden voor een wel toegestane statusovergang zijn hieronder aan de hand van de betreffende C-nummers weergegeven. De ingetekende rechthoeken geven hierbij aan op welke gevallen de genoemde constraints van toepassing zijn. (Hierbij is $t \in v(\text{RU})$, $t' \in v'(\text{RU})$ en $t(\text{RNR}) = t'(\text{RNR})$.)

nieuw / oud	~	GEPL	AANW	VERV	
~	+	+	+	–	C16
GEPL	+	+	+	–	
AANW	–	–	+	+	C17 en C18
VERV	–	–	–	+	C19

Ad C17: $t(\text{OPVL}) = t'(\text{OPVL})$
Ad C18: als $t(\text{SRTR}) \in \{$ 'BHR', 'VPR' $\}$ dan $t(\text{SRTR}) = t'(\text{SRTR})$
Ad C19: $t = t'$.

Dergelijke schematische weergaven kunnen wel eens nuttig zijn tijdens het inventariseren

van dynamische constraints, als hulpmiddel om tot de uiteindelijke formulering te kunnen komen.

Wat betreft het (carrière)verloop van specialisten houdt ons ziekenhuis de volgende regels en nevenvoorwaarden in acht.

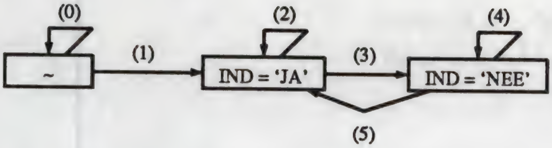
(C20) Elke specialist die in de nieuwe toestand voorkomt als zijnde niet in dienst, komt reeds in de oude toestand voor met dezelfde locum tenens, hetzelfde soort dienstverband, dezelfde specialismen, hetzelfde honorarium per uur, hetzelfde afdelingsnummer en hetzelfde interne telefoonnummer. (Tijdens het uit dienst gaan en het uit dienst zijn van een specialist mogen deze gegevens namelijk niet meer worden veranderd.)

(C21) Een specialist blijft altijd in de administratie aanwezig, met ten minste dezelfde specialismen als voorheen. (Deze eis wordt gesteld omdat er in de gegevens over het medisch handelen, met name bij opnamen, overdrachten, patiëntbehandelingen en medicijnverstrekkingen, veelvuldig wordt verwezen naar de specialisten die er destijds bij waren betrokken.)

(C22) Omdat een specialist altijd in dienst is op basis van een contract met een afdeling, zal een specialist die in dienst is en blijft, in de nieuwe toestand altijd hetzelfde afdelingsnummer, soort dienstverband, aantal uren in de week en aantal toegezegde bedden behouden; verder mag het honorarium per uur niet dalen.

(C23) Bij een transitie zullen de identiteitsnummers voor nieuwe specialisten getallen zijn *vanaf* 100 plus het aantal reeds geadministreerde specialisten; deze nieuwe identiteitsnummers moeten echter wel kleiner zijn dan 100 plus het nieuwe aantal geadministreerde specialisten.

De dynamische constraints voor specialisten, C20 tot en met C23, zullen we weergeven op een soortgelijke manier als die voor ruimten in ons ziekenhuis. We beginnen met een "transitiediagram" op grond van het wel of niet in dienst zijn. In dit diagram representeert geval (1) het in dienst treden, (2) het in dienst blijven, (3) het uit dienst treden, (4) het uit dienst blijven, en (5) het opnieuw in dienst treden.



De "life-cycle" van een specialist

In de nu volgende matrixrepresentatie houden we dezelfde nummering aan als in het voorgaande diagram. Omdat de gegevens van een specialist zijn verdeeld over twee tabellen, de SP-tabel en de MW-tabel, nemen we nu $t \in v(\text{SP}) \bowtie v(\text{MW})$, $t' \in v'(\text{SP}) \bowtie v'(\text{MW})$ en $t(\text{IDNR}) = t'(\text{IDNR})$.

nieuw / oud	~	JA	NEE	
~	(0)	(1)	–	C20
JA	–	(2)	(3)	
NEE	–	(5)	(4)	C21

Ad C20: $t \equiv t'$ op {LOC, SDV, SPEC, HONU, ANR, TELI}; zie Definitie 0.15.
Ad C21: $t(\text{SPEC}) \subseteq t'(\text{SPEC})$.
Ad (2), C22: $t \equiv t'$ op {SDV, AUW, ABD, ANR} en $t(\text{HONU}) \leq t'(\text{HONU})$.

Eis C23, die uitsluitend op geval (1) slaat, is iets anders van aard en kan met behulp van intervallen als volgt worden geformuleerd:

$$\{t'(\text{IDNR}) \mid t' \in v'(\text{SP})\} - \{t(\text{IDNR}) \mid t \in v(\text{SP})\} \subseteq [100 + |v(\text{SP})| \;..\; 100 + |v'(\text{SP})|)$$

We besluiten deze paragraaf nu met de formele definitie van de door de voorgaande dynamische constraints bepaalde transitierelatie RZKH op UZKH.

RZKH =	
$\{(v\,;v') \mid (v\,;v') \in \mathrm{UZKH} \times \mathrm{UZKH}$ en	
$v(\mathrm{AFD}) \upharpoonright \{\mathrm{ANR}\} \subseteq v'(\mathrm{AFD}) \upharpoonright \{\mathrm{ANR}\}$ en	C00
$v(\mathrm{MVS}) \subseteq v'(\mathrm{MVS})$ en	C01
$v(\mathrm{PB}) \subseteq v'(\mathrm{PB})$ en	C02
id$(\{\mathrm{PNR, OPND, RNR, IDNR, NAWH}\})$ verbindt $v(\mathrm{OPN})$ met $v'(\mathrm{OPN})$ en	C03
$v(\mathrm{OPN}) \bowtie v(\mathrm{ONT}) \subseteq v'(\mathrm{OPN}) \bowtie v'(\mathrm{ONT})$ en	C04
$v(\mathrm{OSP}) \not\upharpoonright \{\mathrm{OPNR}\} \subseteq v'(\mathrm{OSP}) \not\upharpoonright \{\mathrm{OPNR}\}$ en	C05
$[\forall t \in v(\mathrm{OVR}): \exists t' \in v'(\mathrm{OVR}): (t \not\upharpoonright \{\mathrm{BEV}\} = t' \not\upharpoonright \{\mathrm{BEV}\}$ en $t(\mathrm{BEV}) \subseteq t'(\mathrm{BEV}))]$ en	C06
$\{\mathrm{MCD}\} \overset{\sim}{\rightarrow} \{\mathrm{MSRT}\}$ bij $(v(\mathrm{MED}); v'(\mathrm{MED}))$ en	C07
$\{\mathrm{BCD}\} \overset{\sim}{\rightarrow} \{\mathrm{BSRT}\}$ bij $(v(\mathrm{BEH}); v'(\mathrm{BEH}))$ en	C08
$\{\mathrm{PNR}\} \overset{\sim}{\rightarrow} \{\mathrm{GPL, GBD, INSD, BLGR, RHF}\}$ bij $(v(\mathrm{PAT}); v'(\mathrm{PAT}))$ en	C09
$[\forall t \in v(\mathrm{PAT}): \forall t' \in v'(\mathrm{PAT}):$ als $t(\mathrm{PNR}) = t'(\mathrm{PNR})$ en $t(\mathrm{GESL}) =$ 'M'	C10
dan $t(\mathrm{NAW})\,(\mathrm{NM}) = t'(\mathrm{NAW})\,(\mathrm{NM})]$ en	,,
$[\forall t \in v(\mathrm{WN}): \forall t' \in v'(\mathrm{WN}):$ als $t(\mathrm{IDNR}) = t'(\mathrm{IDNR})$	
dan $t(\mathrm{SAL})/t(\mathrm{AHD}) \leq t'(\mathrm{SAL})/t'(\mathrm{AHD})]$ en	C11
$[(\forall t \in v(\mathrm{WN}): \forall t' \in v'(\mathrm{WN}):$ als $t(\mathrm{IDNR}) = t'(\mathrm{IDNR})$ dan $t(\mathrm{AZD}) \leq t'(\mathrm{AZD}))$	C12
of $(\forall t' \in v'(\mathrm{WN}): t'(\mathrm{AZD}) = 0)]$ en	,,
$[\forall t \in v(\mathrm{NMP}) \bowtie v(\mathrm{VV}): \forall t' \in v'(\mathrm{NMP}):$ als $t(\mathrm{IDNR}) = t'(\mathrm{IDNR})$	
dan $t(\mathrm{AVD}) \leq t'(\mathrm{AVD})]$ en	C13
$[\forall t \in v(\mathrm{HA}): \forall t' \in v'(\mathrm{HA}):$ als $t(\mathrm{NRHA}) = t'(\mathrm{NRHA})$ en	
t lijkt niet op t' t.a.v. $\{\mathrm{PAT, TOEN, VWIN}\}$	C14
dan $t'(\mathrm{TOEN}) = t'(\mathrm{PAT}) - t(\mathrm{PAT})]$ en	,,
$[\forall t \in v(\mathrm{AFD}): \forall t' \in v'(\mathrm{AFD}):$ als $t(\mathrm{ANR}) = t'(\mathrm{ANR})$	C15
dan $\{t(\mathrm{CHNR}), t(\mathrm{SCNR})\} \cap \{t'(\mathrm{CHNR}), t'(\mathrm{SCNR})\} \neq \emptyset]$ en	,,
id$(\{\mathrm{RNR}\})$ verbindt	
$\{t' \in v'(\mathrm{RU}) \mid t'(\mathrm{STAT}) =$ 'VERV'$\}$ met $\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) \neq$ 'GEPL'$\}$ en	C16
id$(\{\mathrm{RNR, OPVL}\})$ verbindt	
$\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) =$ 'AANW'$\}$ met $\{t' \in v'(\mathrm{RU}) \mid t'(\mathrm{STAT}) \neq$ 'GEPL'$\}$ en	C17
id$(\{\mathrm{RNR, SRTR}\})$ verbindt	
$\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) =$ 'AANW' en $t(\mathrm{SRTR}) \in \{$'BHR', 'VPR'$\}\}$ met $v'(\mathrm{RU})$ en	C18
$\{t \in v(\mathrm{RU}) \mid t(\mathrm{STAT}) =$ 'VERV'$\} \subseteq v'(\mathrm{RU})$ en	C19
id$(\{\mathrm{IDNR, LOC, SDV, SPEC, HONU, ANR, TELI}\})$ verbindt	C20
$\{t' \in v'(\mathrm{SP}) \bowtie v'(\mathrm{MW}) \mid t'(\mathrm{IND}) =$ 'NEE'$\}$ met $v(\mathrm{SP}) \bowtie v(\mathrm{MW})$ en	,,
$[\forall t \in v(\mathrm{SP}) \bowtie v(\mathrm{MW}): \exists t' \in v'(\mathrm{SP}) \bowtie v'(\mathrm{MW}):$	
$(t(\mathrm{IDNR}) = t'(\mathrm{IDNR})$ en $t(\mathrm{SPEC}) \subseteq t'(\mathrm{SPEC})$ en	C21
als $t(\mathrm{IND}) = t'(\mathrm{IND}) =$ 'JA'	C22
dan $(t(\mathrm{HONU}) \leq t'(\mathrm{HONU})$ en t lijkt op t' t.a.v. $\{\mathrm{SDV, AUW, ABD, ANR}\}))]$ en	,,
$\{t'(\mathrm{IDNR}) \mid t' \in v'(\mathrm{SP})\} - \{t(\mathrm{IDNR}) \mid t \in v(\mathrm{SP})\} \subseteq [100 + \mid v(\mathrm{SP}) \mid \,..\, 100 + \mid v'(\mathrm{SP}) \mid)\,\}$.	C23

OPGAVEN

5.15. (a) Is RZKH reflexief op UZKH?

(b) Is RZKH reversibel?

Licht uw antwoorden toe.

5.16. Blijven bij RZKH van specialisten die bij een transitie uit dienst zijn én blijven, alle SP-gegevens ongewijzigd? Zo ja, geef dan aan uit welke constraints dit volgt; zo nee, noem dan *alle* SP-attributen waarvan de waarde in zo'n geval wel kan wijzigen.

5.17. Ga voor elk van de volgende beweringen de geldigheid in UZKH c.q. RZKH na. Noem voor elke geldige bewering de constraint(s) waaruit die bewering volgt; geef voor elke andere bewering aan of het een statische dan wel dynamische constraint is en geef dan tevens de formele weergave.

(a) Een specialist mag nooit van afdeling veranderen.

(b) Zolang een werknemer verplicht verzekerd is, mag die persoon niet van soort afdeling veranderen.

(c) Patiënten die ooit een opname, behandeling of medicijnverstrekking hebben gehad, verdwijnen niet meer uit de administratie.

(d) Behandelingen die ooit uitgevoerd zijn en medicijnen die ooit verstrekt zijn, vervallen niet.

(e) Het aantal in dienst zijnde medewerkers stijgt niet.

(f) Alleen van de meest recente overdracht van een patiënt kan de opnamereden alsnog worden aangepast.

(g) Alleen van de meest recente overdracht van een nog opgenomen patiënt kan de opnamereden alsnog worden aangepast.

5.18. Bedenk zelf nog enige dynamische constraints over UZKH en geef ze zowel in woorden als formeel weer.

5.19. Bewijs dat $\{t'(\mathrm{IDNR}) \mid t' \in v'(\mathrm{SP})\} = [100 \,..\, 100 + \mid v'(\mathrm{SP}) \mid)$ voor elke $v' \in$ UZKH die volgens RZKH bereikbaar is vanuit de lege toestand, dus voor elke $v' \in$ UZKH waarvoor $(\lambda E \in \mathrm{dom}(\mathrm{GZKH}){:}\ \varnothing\,;\ v') \in \mathrm{Tcl}(\mathrm{RZKH})$.

(Hint: gebruik volledige inductie.)

□

6 RAADPLEGING

Onder raadpleging van een database verstaan we, informeel verwoord, het opvragen van gegevens van een willekeurige (DB-)toestand van het (DB-)universum in kwestie. Meer specifiek kunnen we een vraag (of opdracht of "query") opvatten als een functie die aan elke toestand van het betreffende universum een "antwoord" toevoegt, bijvoorbeeld een tabel, een getal of een ja/nee-antwoord. Raadpleging wordt ook wel vaak "retrieval" genoemd.

We maken onderscheid tussen het *formuleren*, het *stellen* en het *uitrekenen* van een vraag.

- Het *formuleren* van een vraag komt neer op het specificeren van een functie over het universum in kwestie, een functie die aan elke toestand het gevraagde toevoegt.
- Het *stellen* van een vraag q aan een database in toestand v komt neer op het toepassen van de functie q op het argument v. We noemen $q(v)$ wel het *antwoord* op vraag q in toestand v.
- Het *uitrekenen* van het antwoord op vraag q in toestand v komt neer op het bepalen van de "meest eenvoudige" representatievorm van het antwoord $q(v)$.

In een geautomatiseerde omgeving is het *formuleren* van vragen typisch een taak voor de ontwerpers, de applicatiebouwers en de meer ervaren gebruikers van het systeem, het *stellen* van vragen vooral een taak van de eindgebruikers, en het *uitrekenen* van vragen typisch een taak voor het (database-management)systeem in kwestie.

In dit hoofdstuk zullen we met name ingaan op het *formuleren* van vragen.

6.1 QUERIES

In de inleiding hebben we reeds informeel verwoord wat we onder een vraag verstaan. We geven nu een formele definitie van dit begrip. In die definitie kan het universum een willekeurige verzameling zijn. Voor dit formele begrip zullen we in het vervolg de Engelse term *query* gebruiken.

DEFINITIE 6.1:
Als U een verzameling is, dan:
q is een **query over** $U \overset{D}{\Longleftrightarrow} q$ is een functie over U.

VOORBEELD 6.1:

We geven twee voorbeelden van queries, één over een tabellenverzameling en één over een DB-universum.

1. De opdracht "Geef het aantal personen die niet in hun geboorteplaats wonen" over de tabellenverzameling WP uit Voorbeeld 2.8 kunnen we formeel als volgt weergeven:

$$\lambda T \in \mathrm{WP} : \mid \{t \in T \mid t(\mathrm{WPL}) \neq t(\mathrm{GPL})\} \mid$$

Het antwoord op deze vraag in de toestand weergegeven in Figuur 2.7 is kennelijk 3.

2. De analoge opdracht "Geef het aantal patiënten die niet in hun geboorteplaats wonen" over het DB-universum UZKH uit Hoofdstuk 4 kunnen we formeel als volgt weergeven:

$$\lambda v \in \mathrm{UZKH} : \mid \{t \in v(\mathrm{PAT}) \mid t(\mathrm{NAW})\,(\mathrm{WPL}) \neq t(\mathrm{GPL})\} \mid$$

□ Voorbeeld 6.1.

Met behulp van de tot nu toe ingevoerde begrippen en notaties zijn we in feite in staat om de meest uiteenlopende queries te specificeren. Om een indruk te geven zullen we in de volgende voorbeelden telkens enige illustratieve klassen van vragen behandelen (K1, K2, etcetera). Daarbij geven we onder (A) de algemene vorm, onder (B) een specifiek voorbeeld (aan de hand van het semantisch rijke DB-universum UZKH uit Hoofdstuk 4), en eventueel onder (C) nog enige opmerkingen. (Overigens doet de informele verwoording bij (B) niet altijd meteen vermoeden dat het in feite een vraag van de onder (A) beschreven vorm betreft.) Enkele vragen onder (B) en in de opgaven zijn varianten op de vragen in [Re 82],

Hoofstuk 8. In de praktijk kan het nuttig zijn om voor een vraag verschillende formele weergaven achter de hand te hebben, bijvoorbeeld omdat een bepaalde op zichzelf correcte oplossing niet zonder meer uitdrukbaar is in de query-taal van het DBMS in kwestie of omdat een andere verwoording opeens tot een aanzienlijk kortere responsetijd kan leiden. We zullen daarom onder (A) soms meer dan één oplossing geven. Onder (B) gebruiken we telkens één van deze genummerde varianten voor het specifieke UZKH-voorbeeld.

In de onder (A) genoemde algemene vormen van de voorbeelden gaan we uit van een DB-universum U over een DB-skelet g, tabelindexen E en E' met $\{a,b,c,d,e\} \subseteq g(E)$, $A \subseteq g(E)$ en $d' \in g(E')$, $B \subseteq g(E) \cap g(E')$, een DB-functie H over U m.b.t. $(E\,;E')$ en een DB-functie G over U m.b.t. $(E'\,;E)$.

Elk van de vragen in Voorbeeld 6.2 heeft betrekking op slechts één tabel.

VOORBEELD 6.2:

K1: (A) Geef alle gegevens uit de E-tabel.

(1) $\lambda v \in U : v(E)$

(B) Geef de gegevens van alle plaatsen in de regio.

(1) $\lambda v \in \text{UZKH} : v(\text{PLR})$

(C) Het antwoord bestaat dus uit een tabel.

K2: (A) Geef de A-waarden van alle E-tupels.

(1) $\lambda v \in U : v(E) \restriction A$

(2) $\lambda v \in U : \{t \restriction A \mid t \in v(E)\}$

(B) Geef code, naam en voorraad van elk medicijn.

(1) $\lambda v \in \text{UZKH} : v(\text{MED}) \restriction \{\text{MCD, MNM, VRD}\}$

(C) Als de attributenverzameling A in (A) geen sleutel van E in U is, dan kan in sommige toestanden v het aantal tupels in het antwoord kleiner zijn dan het aantal tupels in de tabel $v(E)$ zelf. Bijvoorbeeld, bij de vraag "Geef elke medicijnsoort" is A gelijk aan {MSRT} en {MSRT} is geen sleutel van MED in UZKH. In een toestand v waarin er verschillende medicijntupels met dezelfde soort voorkomen zal het antwoord dus uit een kleiner aantal tupels bestaan dan v(MED) zelf.

K3: (A) Geef de A-waarden van elk E-tupel waarvan de a-waarde x of de b-waarde ongelijk aan y is.

(1) $\lambda v \in U : \{t \in v(E) \mid t(a) = x \text{ of } t(b) \neq y\} \upharpoonright A$

(2) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en } (t(a) = x \text{ of } t(b) \neq y)\}$

(3) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en (als } t(b) = y \text{ dan } t(a) = x)\}$

(4) $\lambda v \in U : (\{t \in v(E) \mid t(a) = x\} \cup \{t \in v(E) \mid t(b) \neq y\}) \upharpoonright A$

(5) $\lambda v \in U : (\{t \in v(E) \mid t(a) = x\} \upharpoonright A) \cup (\{t \in v(E) \mid t(b) \neq y\} \upharpoonright A)$

(6) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en } t(a) = x\} \cup$
$\{t \upharpoonright A \mid t \in v(E) \text{ en } t(b) \neq y\}$

(B) Geef patiëntnummer, bloedgroep en rhesusfactor van alle vrouwelijke patiënten en van alle patiënten met kinderen.

(6) $\lambda v \in \text{UZKH}:$
$\{t \upharpoonright \{\text{PNR, BLGR, RHF}\} \mid t \in v(\text{PAT}) \text{ en } t(\text{GESL}) = \text{‘V’}\} \cup$
$\{t \upharpoonright \{\text{PNR, BLGR, RHF}\} \mid t \in v(\text{PAT}) \text{ en } t(\text{KNDG}) \neq \varnothing\}$

(C) De volgende oplossing, waarin t een functie over A is, is alleen correct als $\{a,b\} \subseteq A$.

(7) $\lambda v \in U : \{t \in v(E) \upharpoonright A \mid t(a) = x \text{ of } t(b) \neq y\}$

Zoals aangegeven hebben we in (B) een oplossing conform variant (6) onder (A) gegeven.

K4: (A) Geef het aantal E-tupels waarvan de a-waarde niet als b-waarde in de E-tabel voorkomt.

(1) $\lambda v \in U : | \{t \in v(E) \mid t(a) \notin \{t'(b) \mid t' \in v(E)\}\} |$

(2) $\lambda v \in U : | \{t \in v(E) \mid \forall t' \in v(E) : t'(b) \neq t(a)\} |$

(3) $\lambda v \in U : | \{t \in v(E) \mid 0 = | \{t' \in v(E) \mid t'(b) = t(a)\} | \} |$

(B) Geef het aantal specialisten die geen locum tenens zijn.

(2) $\lambda v \in \text{UZKH} : | \{t \in v(\text{SP}) \mid \forall t' \in v(\text{SP}) : t'(\text{LOC}) \neq t(\text{IDNR})\} |$

(C) Als $\{a\}$ een sleutel van E in U is (zoals bijvoorbeeld in (B) het geval is), dan zijn ook de volgende varianten correct:

(4) $\lambda v \in U : | \{t(a) \mid t \in v(E) \text{ en } t(a) \notin \{t'(b) \mid t' \in v(E)\}\} |$

(5) $\lambda v \in U : | \{t(a) \mid t \in v(E)\} - \{t'(b) \mid t' \in v(E)\} |$

K5: (A) Geef de som van het produkt van de *a*-waarde en de *b*-waarde van alle *E*-tupels waarvan de *c*-waarde x , y of z is.

(1) $\lambda v \in U : (\Sigma t \in v(E) \text{ en } t(c) \in \{x , y , z\} : t(a) * t(b))$

(B) Geef de totale verkoopwaarde van alle medicijnen waarvan de soort 'K', 'N' of 'O' is.

(1) $\lambda v \in \text{UZKH}:$
$(\Sigma t \in v(\text{MED}) \text{ en } t(\text{MSRT}) \in \{ \text{'K'}, \text{'N'}, \text{'O'} \} : t(\text{VRD}) * t(\text{VKPR}))$

(C) Evenals bij de vorige vorm komt hier dus altijd één enkel getal uit.

K6: (A) Geef met 'JA' of 'NEE' aan of er meer dan 1000 *E*-tupels zijn.

(1) $\{(v ; \text{'JA'}) \mid v \in U \text{ en } \mid v(E) \mid > 1000\} \cup$
$\{(v ; \text{'NEE'}) \mid v \in U \text{ en } \mid v(E) \mid \leq 1000\}$

(2) $(\lambda v \in U \text{ en } \mid v(E) \mid > 1000 : \text{'JA'}) \cup$
$(\lambda v \in U \text{ en } \mid v(E) \mid \leq 1000 : \text{'NEE'})$

(3) $(\lambda v \in U : \text{'NEE'})\ \theta\ (\lambda v \in U \text{ en } \mid v(E) \mid > 1000 : \text{'JA'})$

(4) $(\lambda v \in U : \text{'JA'})\ \theta\ (\lambda v \in U \text{ en } \mid v(E) \mid \leq 1000 : \text{'NEE'})$

(B) Geef met 'JA' of 'NEE' aan of er meer dan 1000 regio-artsen zijn.

(3) $(\lambda v \in \text{UZKH} : \text{'NEE'})\ \theta\ (\lambda v \in \text{UZKH en } \mid v(\text{HA}) \mid > 1000 : \text{'JA'})$

(C) Dit is een voorbeeld van een ja/nee-vraag.

K7: (A) Geef van elk *E*-tupel de *a*-waarde, de *b*-waarde en (onder de titel α) het produkt van de *c*-, *d*- en *e*-waarde.

(1) $\lambda v \in U : \{ t \upharpoonright \{a , b\} \cup \{(\alpha; t(c) * t(d) * t(e))\} \mid t \in v(E)\}$

(2) $\lambda v \in U : \{ \{(a ; t(a)), (b ; t(b)), (\alpha; t(c) * t(d) * t(e))\} \mid t \in v(E)\}$

(B) Geef van alle medicijnverstrekkingen het patiëntnummer, de medicijncode en (onder de titel HOEV) de verstrekte hoeveelheid.

(1) $\lambda v \in \text{UZKH}:$
$\{t \upharpoonright \{\text{PNR, MCD}\} \cup \{(\text{HOEV}; t(\text{AEK}) * t(\text{FREQ}) * t(\text{DUUR}))\}$
$\mid t \in v(\text{MVS})\}$

(C) Als voorwaarde voor de keuze van α stellen we dat $\alpha \notin \{a , b\}$.
Onder (A) is het antwoord in elke toestand een tabel over $\{a , b , \alpha\}$.
Als $\{a , b\}$ geen sleutel van E in U is (zoals in het voorbeeld onder (B) het geval is), dan geven de oplossingen per $(a ; b)$-combinatie wel aan *welke* α-waarden er optreden, maar niet *hoe vaak* ze optreden.

K8: (A) Geef elke A-combinatie die in de E-tabel voorkomt en (onder de titel β) het aantal keren dat die combinatie voorkomt.

(1) $\lambda v \in U : \{w \cup \{(\beta;\ |\ \{t \in v(E) \mid t \upharpoonright A = w\}\ |)\} \mid w \in v(E) \boldsymbol{\upharpoonright} A\}$

(B) Geef van elke behandelde patiënt het patiëntnummer en (onder de titel AANTBEH) het aantal keren dat die patiënt is behandeld.

(1) $\lambda v \in \text{UZKH} : \{w \cup \{(\text{AANTBEH};\ |\ \{t \in v(\text{PB}) \mid t \upharpoonright \{\text{PNR}\} = w\}\ |)\}$
$\mid w \in v(\text{PB}) \boldsymbol{\upharpoonright} \{\text{PNR}\}\}$

(C) Onder (A) stellen we als voorwaarde voor de keuze van β dat $\beta \notin A$; het antwoord is in elke toestand een tabel over $A \cup \{\beta\}$.
Wanneer we ook de niet behandelde patiënten genoemd willen zien dan moeten we w niet over $v(\text{PB}) \boldsymbol{\upharpoonright} \{\text{PNR}\}$ maar over $v(\text{PAT}) \boldsymbol{\upharpoonright} \{\text{PNR}\}$ laten "lopen". De vraag heeft dan evenwel betrekking op meer dan één tabel.

□ Voorbeeld 6.2.

Bij de vragen in Voorbeeld 6.3 komen de gegevens weliswaar telkens uit slechts één tabel, maar wel op grond van criteria gebaseerd op andere tabellen.

VOORBEELD 6.3:

K1: (A) Geef de A-waarde van ieder E-tupel waarvan de a-waarde tussen m en n ligt (inclusief m maar exclusief n) en waar ook een E'-tupel met dezelfde B-waarde bijhoort.

(1) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en } m \leq t(a) \text{ en } t(a) < n \text{ en}$
$\exists t' \in v(E') : t' \upharpoonright B = t \upharpoonright B\}$

(2) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en } t(a) \geq m \text{ en } t(a) < n \text{ en}$
$0 < |\ \{t' \in v(E') \mid t' \upharpoonright B = t \upharpoonright B\}\ |\}$

(3) $\lambda v \in U : (\{t \in v(E) \mid t(a) \in [m\, ..\, n)\} \boldsymbol{\upharpoonright} A) \cap$
$(\{t \in v(E) \mid \{t' \in v(E') \mid t' \upharpoonright B = t \upharpoonright B\} \neq \varnothing\} \boldsymbol{\upharpoonright} A)$

(4) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \text{ en } t(a) \in [m\, ..\, n) \text{ en } t \upharpoonright B \in v(E') \boldsymbol{\upharpoonright} B\}$

(5) $\lambda v \in U : ((v(E') \boldsymbol{\upharpoonright} B) \bowtie \{t \in v(E) \mid t(a) \in [m\, ..\, n)\}) \boldsymbol{\upharpoonright} A$

(6) $\lambda v \in U : \{t \upharpoonright A \mid t \in v(E) \bowtie (v(E') \boldsymbol{\upharpoonright} B) \text{ en } t(a) \in [m\, ..\, n)\}$

(B) Geef patiëntnummer, geboortedatum, inschrijfdatum en NAW-gegevens van alle in 1989 geboren patiëntjes die opgenomen zijn of opgenomen zijn geweest.

(5) $\lambda v \in \text{UZKH} : ((v(\text{OPN}) \upharpoonright \{\text{PNR}\}) \bowtie \{t \in v(\text{PAT}) \mid t(\text{GBD}) \in [19890101 \,..\, 19900101)\}) \upharpoonright \{\text{PNR, GBD, INSD, NAW}\}$

K2: (A) Geef onder de titels α, β en γ achtereenvolgens de *a*-, *b*- en *c*-waarde van elk *E*-tupel waarbij de *d'*-waarde van het bijbehorende *E'*-tupel volgens de DB-functie *H* groter dan *w* is.

(1) $\lambda v \in U : \{\{(\alpha; t(a)), (\beta; t(b)), (\gamma; t(c))\} \mid t \in v(E) \text{ en } H(v)\,(t)\,(d') > w\}$

(B) Geef onder de titels PATIENT, DATUM en LOCATIE achtereenvolgens het patiëntnummer, de datum van overplaatsing en het nieuwe verpleegruimtenummer van iedere overplaatsing gedurende een opname die na 10 juni 1989 begon.

(1)
$$\begin{aligned}\lambda v \in \text{UZKH} : \{\{&(\text{PATIENT}\,; t(\text{PNR})),\\ &(\text{DATUM}\;\;; t(\text{BDAT})),\\ &(\text{LOCATIE}; t(\text{RNR}))\}\\ &\mid t \in v(\text{OVR}) \text{ en Hovropn}(v)\,(t)\,(\text{OPND}) > 890610\}\end{aligned}$$

(C) We zien hier dus hoe we de heading van een resultaattabel zelf kunnen "verfraaien".

Wanneer we de in §4.3 gegeven definitie van de DB-functie Hovropn uitschrijven, dan komen we tot de volgende alternatieve formulering:

$$\begin{aligned}\lambda v \in \text{UZKH} : \{\{&(\text{PATIENT}\,; t(\text{PNR})),\\ &(\text{DATUM}\;\;; t(\text{BDAT})),\\ &(\text{LOCATIE}; t(\text{RNR}))\}\\ &\mid t \in v(\text{OVR}) \text{ en}\\ &\;\exists u \in v(\text{OPN}) \text{: } (t(\text{PNR}) = u(\text{PNR}) \text{ en}\\ &\qquad t(\text{BDAT}) > u(\text{OPND}) \text{ en}\\ &\qquad u(\text{OPND}) > 890610 \text{ en}\\ &\qquad \text{als } u(\text{ONTS}) = \text{`JA'}\\ &\qquad \text{dan } t(\text{BDAT}) \leq u(\text{ONTD}))\}\end{aligned}$$

We zien hier duidelijk het nut van DB-functies bij het formuleren van queries. Dit is op zich niet zo verwonderlijk wanneer we bedenken dat DB-functies de formaliseringen zijn van de intuïtieve verbanden die er tussen de tabellen geacht worden te bestaan.

K3: (A) Geef de E-tupels die volgens de gegeneraliseerde compositie $G \odot H$ naar zichzelf verwijzen (dus via de E'-tabel).

(1) $\lambda v \in U : \{t \in v(E) \mid (G \odot H)(v)(t) = t\}$

(2) $\lambda v \in U : \{t \in v(E) \mid (G(v) \circ H(v))(t) = t\}$

(3) $\lambda v \in U : \{t \in v(E) \mid G(v)(H(v)(t)) = t\}$

(B) Geef de gegevens van alle afdelingen waarvan de chef een medewerker van diezelfde afdeling is.

(1) $\lambda v \in \text{UZKH} : \{t \in v(\text{AFD}) \mid (\text{Hmwafd} \odot \text{Hafdmw})(v)(t) = t\}$

(C) Onder (A) vragen we voor elke toestand v dus naar de *dekpunten* van de functie $(G \odot H)(v)$.
We geven voor (B) ook nog een oplossing zonder DB-functies:

$$\lambda v \in \text{UZKH} : \{t \in v(\text{AFD}) \mid \exists t' \in v(\text{MW}) : (t'(\text{IDNR}) = t(\text{CHNR}) \text{ en } t'(\text{ANR}) = t(\text{ANR}))\}$$

□ Voorbeeld 6.3.

In het volgende voorbeeld behandelen we enige vragen die met name ingaan op de situatie dat er een DB-functie G over U met betrekking tot het tabelindexpaar $(E' ; E)$ gegeven is. De vragen in Voorbeeld 6.4 zijn vaak kleine variaties van elkaar. We zullen hierbij zien hoe sommige oplossingen zich wel en sommige zich niet goed lenen voor simpele aanpassingen bij zulke kleine variaties. Deze aanpassingsproblemen bij kleine wijzigingen in de vraagstelling staan model voor die welke we bij talen als SQL kunnen tegenkomen.

Voor de lezer die bekend is met CODASYL (zie bijvoorbeeld [CO 71], [Ol 78] of [Re 82]) merken we op dat bij een gedisciplineerde aanpak in een klassieke CODASYL-omgeving de DB-functie G met behulp van een (AUTOMATIC MANDATORY) "DBTG set" kan worden geïmplementeerd. Hierbij dient dan wel in elke toestand $v \in U$ voor elke $t' \in v(E')$ als "owner" van t' het E-tupel $G(v)(t')$ te worden gekozen. In dat geval representeert voor $t \in v(E)$ de tupelverzameling $G(v)$ inv t de verzameling "members" van t. In een dergelijke 3^{e} generatie-omgeving kunnen onze formele queries nu dienst doen als de *formele specificaties* van de te schrijven (COBOL-)programma's.

In Voorbeeld 6.4 geven we onder (C) enige additionele oplossingen in het geval dat

$$\forall v \in U : \forall t \in v(E) : \forall t' \in v(E') : (G(v)(t') = t \Leftrightarrow t' \text{ en } t \text{ zijn compatibel}) , \qquad (**)$$

een niet ongebruikelijke situatie in de praktijk. Zo geeft bijvoorbeeld Opgave 2.86 enige voldoende voorwaarden opdat (**) geldt. We wijzen er verder op dat t' en t compatibel zijn desda $t' \upharpoonright g(E) = t \upharpoonright g(E')$.

VOORBEELD 6.4:

K1: (A) Geef alle E-tupels waar volgens G een E'-tupel bijhoort.

(1) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in v(E') : G(v)\,(t') = t\}$

(2) $\lambda v \in U : \{t \in v(E) \mid \{t' \in v(E') \mid G(v)\,(t') = t\} \neq \varnothing\}$

(3) $\lambda v \in U : \{t \in v(E) \mid (G(v) \operatorname{inv} t) \neq \varnothing\}$

(4) $\lambda v \in U : \{t \in v(E) \mid \; \mid G(v) \operatorname{inv} t \mid \neq 0\}$

(5) $\lambda v \in U : \{t \in v(E) \mid \; \mid G(v) \operatorname{inv} t \mid \geq 1\}$

(B) Geef alle patiënten die wel eens zijn behandeld.

(3) $\lambda v \in \text{UZKH} : \{t \in v(\text{PAT}) \mid (\text{Hpbpat}(v) \operatorname{inv} t) \neq \varnothing\}$

(C) Als (**) geldt dan zijn ook de volgende oplossingen correct:

(6) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in v(E') : t' \upharpoonright g(E) = t \upharpoonright g(E')\}$

(7) $\lambda v \in U : (v(E) \bowtie v(E')) \Vert\!\upharpoonright g(E)$

(8) $\lambda v \in U : v(E) \bowtie (v(E') \Vert\!\upharpoonright g(E))$

Uitwerking van oplossing (6) voor de vraag onder (B) levert als resultaat:

(6) $\lambda v \in \text{UZKH} : \{t \in v(\text{PAT}) \mid \exists t' \in v(\text{PB}) : t'(\text{PNR}) = t(\text{PNR})\}$

Terzijde merken we op dat de eerste vijf oplossingen het *functionele verband* tussen de tabellen benadrukken, oplossing (6) in de stijl van de *relational calculus* is geschreven, en de oplossingen (7) en (8) binnen het kader van de *relational algebra* passen. Voor nadere uitleg van deze drie termen verwijzen we de lezer naar [Sh 81] en [Co 72b].

K2: (A) Geef alle E-tupels waar volgens G geen E'-tupel bijhoort.

(1) $\lambda v \in U : \{t \in v(E) \mid \forall t' \in v(E') : G(v)\,(t') \neq t\}$

(2) $\lambda v \in U : \{t \in v(E) \mid \{t' \in v(E') \mid G(v)\,(t') = t\} = \varnothing\}$

(3) $\lambda v \in U : \{t \in v(E) \mid (G(v) \operatorname{inv} t) = \varnothing\}$

(4) $\lambda v \in U : \{t \in v(E) \mid \; \mid G(v) \operatorname{inv} t \mid = 0\}$

(5) $\lambda v \in U : \{t \in v(E) \mid \; \mid G(v) \operatorname{inv} t \mid < 1\}$

(B) Geef alle behandelingen die nooit zijn uitgevoerd.

(4) $\lambda v \in \text{UZKH} : \{t \in v(\text{BEH}) \mid \mid \text{Hpbbeh}(v) \text{ inv } t \mid = 0\}$

(C) Als (**) geldt dan zijn ook de volgende oplossingen correct:

(6) $\lambda v \in U : \{t \in v(E) \mid \forall t' \in v(E') : t' \restriction g(E) \neq t \restriction g(E')\}$

(7) $\lambda v \in U : v(E) - (v(E) \bowtie v(E')) \restriction g(E)$

(8) $\lambda v \in U : v(E) - (v(E) \bowtie (v(E') \restriction g(E)))$

K3: (A) Geef alle E-tupels waar volgens G ten minste drie E'-tupels bijhoren.

(1) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in v(E') : \exists t'' \in v(E') : \exists t''' \in v(E') :$
$G(v)\,(t') = t \text{ en } G(v)\,(t'') = t \text{ en } G(v)\,(t''') = t$
$\text{en } t' \neq t'' \text{ en } t' \neq t''' \text{ en } t'' \neq t'''\}$

(2) $\lambda v \in U : \{t \in v(E) \mid \mid G(v) \text{ inv } t \mid \geq 3\}$

(B) Geef de gegevens van alle afdelingen met ten minste drie medewerkers.

(2) $\lambda v \in \text{UZKH} : \{t \in v(\text{AFD}) \mid \mid \text{Hmwafd}(v) \text{ inv } t \mid \geq 3\}$

(C) Het analogon van oplossing (6) onder K1 in het geval dat (**) geldt, wordt (in Opgave 6.3) aan de lezer overgelaten.
De oplossingen (7) en (8) onder K1 lenen zich niet voor rechtstreekse aanpassing aan de vraagstelling onder K3.

K4: (A) Geef alle E-tupels waar volgens G een E'-tupel met d'-waarde w' bijhoort.

(1) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in (G(v) \text{ inv } t) : t'(d') = w'\}$

(2) $\lambda v \in U : \{t \in v(E) \mid \{t' \in (G(v) \text{ inv } t) \mid t'(d') = w'\} \neq \varnothing\}$

(3) $\lambda v \in U : \{t \in v(E) \mid \mid \{t' \in (G(v) \text{ inv } t) \mid t'(d') = w'\} \mid \geq 1\}$

(4) $\lambda v \in U : \{t \in v(E) \mid w' \in \{t'(d') \mid t' \in (G(v) \text{ inv } t)\}\}$

(B) Geef de gegevens van alle opnamen gedurende welke er een overdracht aan specialist 129 plaatsvond.

(1) $\lambda v \in \text{UZKH} : \{t \in v(\text{OPN}) \mid \exists t' \in (\text{Hospopn}(v) \text{ inv } t) : t'(\text{IDNR}) = 129\}$

(C) Als (**) geldt dan zijn bijvoorbeeld ook de volgende oplossingen correct:

(5) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in v(E') : (t' \restriction g(E) = t \restriction g(E') \text{ en } t'(d') = w')\}$

(6) $\lambda v \in U : (v(E) \bowtie \{t' \in v(E') \mid t'(d') = w'\}) \restriction g(E)$

(7) $\lambda v \in U : \{t' \in v(E) \bowtie v(E') \mid t'(d') = w'\} \restriction g(E)$

Voor de DB-functie Hospopn in ons voorbeeld onder (B) geldt (**) niet. Uitwerking van deze DB-functie levert het volgende resultaat op (zie de definitie in §4.3):

$$\begin{aligned}\lambda v \in \text{UZKH} : \{t \in v(\text{OPN}) \mid \exists t' \in v(\text{OSP}) : (&t'(\text{IDNR}) = 129 \text{ en} \\ &t'(\text{PNR}) = t(\text{PNR}) \text{ en} \\ &t(\text{OPND}) < t'(\text{BDAT}) \text{ en} \\ &\text{als } t(\text{ONTS}) = \text{'JA'} \\ &\text{dan } t'(\text{BDAT}) \leq t(\text{ONTD}))\}\end{aligned}$$

K5: (A) Geef alle E-tupels waar volgens G een E'-tupel met een andere d'-waarde dan w' bijhoort.

(1) $\lambda v \in U : \{t \in v(E) \mid \exists t' \in (G(v) \operatorname{inv} t) : t'(d') \neq w'\}$

(2) $\lambda v \in U : \{t \in v(E) \mid \{t' \in (G(v) \operatorname{inv} t) \mid t'(d') \neq w'\} \neq \varnothing\}$

(3) $\lambda v \in U : \{t \in v(E) \mid (\{t'(d') \mid t' \in (G(v) \operatorname{inv} t)\} - \{w'\}) \neq \varnothing\}$

(B) Geef de SP-gegevens van elke specialist die wel eens wat anders dan een medicijn met code 'ASPRO' heeft voorgeschreven.

(2) $\lambda v \in \text{UZKH} : \{t \in v(\text{SP}) \mid \{t' \in (\text{Hmvssp}(v) \operatorname{inv} t) \mid t'(\text{MCD}) \neq \text{'ASPRO'}\} \neq \varnothing\}$

(C) Als (**) geldt dan zijn bijvoorbeeld ook de volgende oplossingen correct:

(4) $\lambda v \in U : (v(E) \bowtie \{t' \in v(E') \mid t'(d') \neq w'\}) \upharpoonright g(E)$

(5) $\lambda v \in U : \{t' \in v(E) \bowtie v(E') \mid t'(d') \neq w'\} \upharpoonright g(E)$

De ervaring heeft ons geleerd dat er op de vraag "Geef alle E-tupels waar volgens G geen E'-tupel met d'-waarde w' bijhoort" (ten onrechte) vaak een oplossing behorende bij de klasse K5 wordt gegeven. In Opgave 6.4 komen we op deze vraag terug.

K6: (A) Geef alle E-tupels waar volgens G in de E'-tabel wel de d'-waarden x, x' en x'' maar niet de d'-waarden y en y' bijhoren.

(1) $\lambda v \in U : \{t \in v(E) \mid \{t'(d') \mid t' \in (G(v) \operatorname{inv} t)\} \cap \{x, x', x'', y, y'\} = \{x, x', x''\}\}$

(B) Geef de gegevens van elke patiënt die wel de behandelingen K1, K3 en K5 maar niet de behandelingen K2 en K4 heeft ondergaan.

(1) $\lambda v \in$ UZKH : $\{t \in v(\text{PAT}) \mid \{t'(\text{BCD}) \mid t' \in (\text{Hpbpat}(v) \text{ inv } t)\} \cap$
$\{\text{'K1'}, \text{'K2'}, \text{'K3'}, \text{'K4'}, \text{'K5'}\} = \{\text{'K1'}, \text{'K3'}, \text{'K5'}\}\}$

□ Voorbeeld 6.4.

Bij de vragen in Voorbeeld 6.5 zijn de gegevens in het antwoord afkomstig van verschillende tabellen tegelijk. Gezien het ad hoc karakter van zulke vragen zullen we in Voorbeeld 6.5 geen algemene vormen geven maar ons beperken tot specifieke vragen over UZKH (V1, V2, etcetera).

VOORBEELD 6.5:

V1: Geef alle persoonsgegevens van alle verplicht verzekerde leden van het niet-medisch personeel. (Onder "alle persoonsgegevens" verstaan we in dit geval alle MW-, WN-, NMP- en VV-gegevens.)

(1) $\lambda v \in \text{UZKH} : v(\text{MW}) \bowtie v(\text{WN}) \bowtie v(\text{NMP}) \bowtie v(\text{VV})$

Bij *differentiatie* en *generalisatie* - zie Hoofdstuk 3 - raken de gegevens van één "object" gewoonlijk verspreid over diverse tabellen. Reconstructie van al die verspreide objectgegevens kan meestal geschieden met behulp van joins (zoals geïllustreerd in dit voorbeeld).

V2: Geef van elke behandeling op 10 november 1986 waarbij noch de behandelende specialist, noch de assistent-specialist is verbonden aan de afdeling waarin de behandeling plaatsvond: behandelcode en volgnummer en van de patiënt het nummer, de NAW-gegevens en de geboortedatum.

(1) $\lambda v \in \text{UZKH} : \{t \upharpoonright \{\text{BCD}, \text{VNR}\} \cup \text{Hpbpat}(v)(t) \upharpoonright \{\text{PNR}, \text{GBD}, \text{NAW}\}$
$\mid t \in v(\text{PB})$ en $t(\text{BDAT}) = 861110$ en
$(\text{Hspmw} \odot \text{Hpbsp})(v)(t)(\text{ANR}) \neq \text{Hpbru}(v)(t)(\text{ANR})$ en
$(\text{Hspmw} \odot \text{Hpbspa})(v)(t)(\text{ANR}) \neq \text{Hpbru}(v)(t)(\text{ANR})\}$

(2) $\lambda v \in \text{UZKH} : \{t \upharpoonright \{\text{BCD}, \text{VNR}\} \cup t' \upharpoonright \{\text{PNR}, \text{GBD}, \text{NAW}\}$
$\mid (t\,;t') \in v(\text{PB}) \times v(\text{PAT})$ en
$t(\text{PNR}) = t'(\text{PNR})$ en $t(\text{BDAT}) = 861110$ en
$\exists r \in v(\text{RU}) : \exists m \in v(\text{MW}) : \exists m' \in v(\text{MW}) :$
$[r(\text{RNR}) = t(\text{RNR})$ en $m(\text{IDNR}) = t(\text{IDNR})$ en
$m'(\text{IDNR}) = t(\text{ASNR})$ en $m(\text{ANR}) \neq r(\text{ANR})$ en

$$m'(\text{ANR}) \neq r(\text{ANR})]\}$$

(3) $\lambda v \in \text{UZKH} : \{t \mid t \in v(\text{PB}) \bowtie v(\text{PAT}) \text{ en } t(\text{BDAT}) = 861110 \text{ en}$
$\forall r \in v(\text{RU}) : \forall m \in v(\text{MW}) : \forall m' \in v(\text{MW}) :$
$[\text{ als } r(\text{RNR}) = t(\text{RNR}) \text{ en } m(\text{IDNR}) = t(\text{IDNR}) \text{ en}$
$m'(\text{IDNR}) = t(\text{ASNR})$
$\text{dan } m(\text{ANR}) \neq r(\text{ANR}) \text{ en } m'(\text{ANR}) \neq r(\text{ANR})]$
$\} \upharpoonright \{\text{BCD, VNR, PNR, GBD, NAW}\}$

Bij oplossing (1) maken we dankbaar gebruik van de in §4.3 gedefinieerde DB-functies; oplossing (2) staat echter dichter bij een mogelijke oplossing in SQL.

De "∃-oplossing" onder (2) en de "∀-oplossing" onder (3) zijn equivalent omdat {RNR} en {IDNR} sleutels van RU respectievelijk MW in UZKH zijn!

V3: Geef onder de titels AANVANG, NUMMER, NAAM en AANTAL van alle opnamen die in juni 1989 eindigden achtereenvolgens de aanvangsdatum, het nummer en de naam van de betreffende patiënt, en het aantal verschillende afdelingen waarop de betreffende patiënt gedurende die opname heeft gelegen.

(1) $\lambda v \in \text{UZKH} :$
$\{\{(\text{AANVANG} ; t(\text{OPND})),$
$(\text{NUMMER} ; t(\text{PNR})),$
$(\text{NAAM} ; \text{Hopnpat}(v)\,(t)\,(\text{NAW})\,(\text{NM})),$
$(\text{AANTAL} ; \mid \{\text{Hopnru}(v)\,(t)\,(\text{ANR})\} \cup$
$\{\text{Hovrru}(v)\,(u)\,(\text{ANR}) \mid u \in \text{Hovropn}(v) \text{ inv } t\} \mid)\}$
$\mid t \in v(\text{OPN}) \text{ en } t(\text{ONTD}) \text{ div } 100 = 8906 \text{ en } t(\text{ONTS}) = \text{'JA'}\}$

(2) $\lambda v \in \text{UZKH} :$
$\{\{(\text{AANVANG} ; t(\text{OPND})),$
$(\text{NUMMER} ; t(\text{PNR})),$
$(\text{NAAM} ; t'(\text{NAW})\,(\text{NM})),$
$(\text{AANTAL} ; \mid \{r(\text{ANR}) \mid r \in v(\text{RU}) \text{ en } [t(\text{RNR}) = r(\text{RNR}) \text{ of}$
$\exists u \in (\text{Hovropn}(v) \text{ inv } t) : u(\text{RNR}) = r(\text{RNR})]\} \mid)\}$
$\mid (t ; t') \in v(\text{OPN}) \times v(\text{PAT}) \text{ en } t(\text{PNR}) = t'(\text{PNR}) \text{ en}$
$890601 \leq t(\text{ONTD}) \text{ en } t(\text{ONTD}) < 890701 \text{ en } t(\text{ONTS}) = \text{'JA'}\}$

Evenals bij oplossing (3) van vraag V2 is ook hier een oplossing met behulp van een join (in plaats van een cartesisch produkt) mogelijk. Daarnaast zouden we de DB-functie Hovropn in oplossing (2) nog kunnen wegwerken. We verwijzen in dit verband naar Opgave 6.8.

V4: Geef het zogeheten verpleegkundige verslag van de (eventuele) opname van patiënt 123210 begonnen op 10 juli 1989. (Onder het *verpleegkundige verslag* van een opname verstaan we in ons ziekenhuis van Hoofdstuk 4 een tabel over {VANAF, BEVINDINGEN} die van elke periode waarin de opname door eventuele overplaatsingen verdeeld is, de ingangsdatum en de verpleegkundige bevindingen tijdens die periode bevat.)

(1) $\lambda v \in$ UZKH :

$$\begin{aligned}&\{\{(\text{VANAF}; t(\text{OPND})), (\text{BEVINDINGEN}; t(\text{BEV}))\}\\ &\mid t \in v(\text{OPN}) \text{ en } t(\text{PNR}) = 123210 \text{ en } t(\text{OPND}) = 890710\} \cup\\ &\{\{(\text{VANAF}; t(\text{BDAT})), (\text{BEVINDINGEN}; t(\text{BEV}))\}\\ &\mid t \in v(\text{OVR}) \text{ en } t(\text{PNR}) = 123210 \text{ en}\\ &\ \text{Hovropn}(v)\,(t)\,(\text{OPND}) = 890710\}\end{aligned}$$

Als er geen opname van patiënt 123210 op 10 juli 1989 bestaat in toestand v, dan is volgens bovenstaande definitie het verpleegkundige verslag kennelijk de lege tabel (ongeacht de vraag of er eigenlijk wel een patiënt met nummer 123210 bestaat). Als zo'n opname wel bestaat en er n overplaatsingen zijn (met $n \geq 0$), dan is het aantal tupels in het verpleegkundige verslag kennelijk $n + 1$.

V5: In ons ziekenhuis noemen we een patiëntbehandeling(stupel) x een *directe aangever* van eventuele besmettingen voor een patiëntbehandeling(stupel) y dan en slechts dan als x vóór of op dezelfde dag als y plaatsvindt en x met y de behandelde patiënt of iemand van het behandelteam (dat wil zeggen behandelend specialist plus eventuele assistent-specialist) gemeen heeft. In het bijzonder is elk patiëntbehandelingstupel een directe aangever voor zichzelf.
We definiëren nu voor elke $v \in$ UZKH de relatie DA_v als de verzameling geordende paren $(x\,;y)$ waarvoor x in toestand v een directe aangever voor y is:

$$\begin{aligned}DA_v = \{(x\,;y) \mid\ & (x\,;y) \in v(\text{PB}) \times v(\text{PB}) \text{ en } x(\text{BDAT}) \leq y(\text{BDAT}) \text{ en}\\ & [x(\text{PNR}) = y(\text{PNR}) \text{ of}\\ & \{x(\text{IDNR}), x(\text{ASNR})\} \cap \{y(\text{IDNR}), y(\text{ASNR})\} \neq \varnothing]\}\end{aligned}$$

Merk op dat DA_v *reflexief* is, dat wil zeggen dat $\text{id}(v(\text{PB})) \subseteq DA_v$.

Door een ongelukje op 8 september 1988 zijn mogelijk alle specialisten van afdeling 9 besmet geraakt. We willen daarom van elke patiënt die als gevolg daarvan op directe of indirecte wijze via een behandeling besmet kan zijn geraakt, behalve het patiëntnummer, de NAW-gegevens en de bloedgroep ook het identiteitsnummer en de naam van de contactpersoon van de woonplaats van die patiënt weten.

We kunnen deze query over UZKH, die *recursief* van aard is, als volgt formeel weergeven:

$$\begin{aligned}
&\lambda v \in \mathrm{UZKH}: \\
&\quad \{t \restriction \{\mathrm{PNR}, \mathrm{NAW}, \mathrm{BLGR}\} \cup \\
&\quad\ \{(\mathrm{CPRS}; t'(\mathrm{IDNR})), (\mathrm{NAAM}; t'(\mathrm{NAW})(\mathrm{NM}))\} \\
&\quad \mid (t; t') \in v(\mathrm{PAT}) \times v(\mathrm{MW}) \text{ en} \\
&\quad\ t'(\mathrm{IDNR}) = \mathrm{Hpatplr}(v)(t)(\mathrm{CPRS}) \text{ en} \\
&\quad\ \exists (x; y) \in \mathrm{Tcl}(\mathrm{DA}_v) : (y(\mathrm{PNR}) = t(\mathrm{PNR}) \text{ en } x(\mathrm{BDAT}) \geq 880908 \text{ en} \\
&\quad \{x(\mathrm{IDNR}), x(\mathrm{ASNR})\} \cap \{m(\mathrm{IDNR}) \mid m \in v(\mathrm{MW}) \text{ en } m(\mathrm{ANR}) = 9\} \neq \varnothing)\}
\end{aligned}$$

We hebben bij deze recursieve query gebruik gemaakt van $\mathrm{Tcl}(\mathrm{DA}_v)$, de transitieve afsluiting van de relatie DA_v; zie Definitie 0.22(a).

□ Voorbeeld 6.5.

In Voorbeeld 6.6 gaan we in op vragen waarvan het antwoord gewoonlijk uit meer dan alleen maar één enkele tabel bestaat. Dergelijke vragen kunnen we tegenkomen in de sfeer van (week)overzichten, (maand)rapporten, (jaar)verslagen, dossiers en database-dumps bijvoorbeeld. Ook zogeheten *external schema's* of *subschema's* kunnen we op deze wijze formeel behandelen.

VOORBEELD 6.6:

Onder K2 maken we gebruik van het begrip "deelskelet". We noemen g' een *deelskelet* van het DB-skelet g desda g' een verzamelingsfunctie is en $\mathrm{dom}(g') \subseteq \mathrm{dom}(g)$ en $\forall E \in \mathrm{dom}(g') : g'(E) \subseteq g(E)$. Dus een deelskelet van g is een DB-skelet waarvan elke tabelindex tevens een tabelindex van g is, en wel zodanig dat alle attributen van die tabelindex volgens dat deelskelet ook attributen van die tabelindex volgens g zijn.

Als voorbeeld definiëren we het voor de afdeling Personeelszaken relevante deelskelet PZS van het in Hoofdstuk 4 gebruikte DB-skelet GZKH als volgt:

$$\begin{aligned}
\mathrm{PZS} = {} & \mathrm{GZKH} \restriction \{\mathrm{MW}, \mathrm{WN}, \mathrm{MP}, \mathrm{NMP}, \mathrm{VV}\} \\
& \cup \{(\mathrm{SP}; \mathrm{GZKH}(\mathrm{SP}) - \{\mathrm{LOC}, \mathrm{ABD}\})\}
\end{aligned}$$

Dus het deelskelet PZS bestaat uit de tabelindexen MW, WN, MP, NMP, VV en SP met haast alle bijbehorende attributen; alleen de attributen LOC en ABD van de tabelindex SP doen niet mee.

K1: (A) Geef de gehele database-toestand. (Dit wordt wel een *dump* van de database genoemd.)

(1) $\lambda v \in U : v$

(2) $\mathrm{id}(U)$

(B) Geef alle gegevens in onze ziekenhuis-database.

(2) $\mathrm{id}(\mathrm{UZKH})$

(C) Een praktische toepassing van deze overigens vrij triviale query vormt het maken van "back-ups".

K2: (A) Geef alle gegevens volgens het deelskelet g', het zogenaamde *subschema* volgens g'.

(1) $\lambda v \in U : (\lambda E \in \mathrm{dom}(g') : v(E) \mathbin{\upharpoonright\!\upharpoonright} g'(E))$

(B) Geef alle gegevens volgens het deelskelet PZS.

$$\lambda v \in \mathrm{UZKH} : v \upharpoonright \{\mathrm{MW}, \mathrm{WN}, \mathrm{MP}, \mathrm{NMP}, \mathrm{VV}\}$$
$$\cup \{(\mathrm{SP}; v(\mathrm{SP}) \mathbin{\not\upharpoonright\!\upharpoonright} \{\mathrm{LOC}, \mathrm{ABD}\})\}$$

(C) Als $g'(E) = g(E)$ voor alle E in $\mathrm{dom}(g')$, dan kunnen we de oplossing onder (A) vereenvoudigen tot

(2) $\lambda v \in U : v \upharpoonright \mathrm{dom}(g')$

Onze oplossing onder (B) is mede gebaseerd op bovenstaande eigenschap.

K3: (A) Geef voor alle tabelindexen met het attribuut a alle tupels waarvan de a-waarde w is (exclusief die a-kolom zelf).

(1) $$\lambda v \in U : \{(E ; \{t \mathbin{\not\upharpoonright} \{a\} \mid t \in v(E) \text{ en } t(a) = w\})$$
$$\mid E \in \mathrm{dom}(g) \text{ en } a \in g(E)\}$$

(2) $\lambda v \in U : (\lambda E \in \mathrm{dom}(g) \text{ en } a \in g(E) : \{t \in v(E) \mid t(a) = w\} \mathbin{\not\upharpoonright} \{a\})$

(B) Geef alle persoonlijke en medische gegevens van patiënt 177912 (exclusief het patiëntnummer zelf).

(1) $$\lambda v \in \mathrm{UZKH} : \{(E ; \{t \mathbin{\not\upharpoonright} \{\mathrm{PNR}\} \mid t \in v(E) \text{ en } t(\mathrm{PNR}) = 177912\})$$
$$\mid E \in \{\mathrm{PAT}, \mathrm{MVS}, \mathrm{PB}, \mathrm{OPN}, \mathrm{ONT}, \mathrm{OSP}, \mathrm{OVR}\}\}$$

(C) Merk op dat het antwoord bij K3 altijd een DB-toestand is over het deelskelet

$$\{(E\,;g(E)-\{a\}) \mid E \in \mathrm{dom}(g) \text{ en } a \in g(E)\}.$$

Desondanks is K3 echter geen speciaal geval van K2, en wel omdat bij K2 alle tupels meedoen terwijl er bij K3 ook nog een selectie plaatsvindt.

K4: (A) Geef in een tabel over {ENTITEIT, AANTAL} een overzicht van alle tabelindexen met het bijbehorende aantal elementen.

(1) $\lambda v \in U : \{\{(\mathrm{ENTITEIT}\,; E), (\mathrm{AANTAL}\,; \mid v(E) \mid)\} \mid E \in \mathrm{dom}(g)\}$

(B) Geef in een tabel over {ENTITEIT, AANTAL} een overzicht van alle tabelindexen met het bijbehorende aantal elementen in onze ziekenhuisdatabase.

(1) $\lambda v \in \mathrm{UZKH} : \{\{(\mathrm{ENTITEIT}\,; E), (\mathrm{AANTAL}\,; \mid v(E) \mid)\}$
$\qquad \mid E \in \mathrm{dom}(\mathrm{GZKH})\}$

(C) Merk op dat deze vraag deels over het DB-skelet (de "structuur") en deels over de DB-toestand (de "inhoud") gaat.

□ Voorbeeld 6.6.

Met de voorgaande, gevarieerde selectie van voorbeelden hebben we een indruk willen geven van de vele soorten van vragen die gesteld kunnen worden en van de manier(en) waarop die vragen formeel gespecificeerd kunnen worden.

Wat bij onze voorbeelden misschien ook al enigszins naar voren komt en in de praktijk meestal in nog veel sterkere mate het geval zal zijn, is dat zelfs bij zorgvuldige formulering van een vraag in natuurlijke taal de precieze betekenis van de vraag toch nog niet altijd duidelijk is. Ook is het niet ongebruikelijk dat een vraag bij nader inzien zelfs voor meer dan één uitleg vatbaar is. Bovendien laat de zorgvuldigheid van de formulering van de vraag zelf ook vaak nog te wensen over. In die gevallen is nader overleg met de vragensteller dan ook onontbeerlijk.

OPGAVEN

6.1. Geef de volgende vragen en opdrachten over het DB-universum UZKH uit Hoofdstuk 4 formeel weer.

(a) Geef nummer en naam van elke afdeling waarin in 1987 geen patiëntes uit Waalre werden behandeld.

(b) Zijn er afdelingen waarin in 1987 geen patiëntes uit Waalre werden behandeld?

(c) Is het waar dat er meer niet-medisch dan medisch personeel is?

(d) Geef nummer en naam van alle specialisten die chef zijn van de afdeling waaraan ze formeel verbonden zijn en bovendien maximaal drie verschillende *soorten* behandelingen hebben verricht (als behandelend specialist dan wel als assistent-specialist).

(e) Geef alle persoonsgegevens van alle contactpersonen die minstens twee plaatsen vertegenwoordigen waar meer dan 100 ingeschreven patiënten wonen. (Onder "alle persoonsgegevens" verstaan we in dit geval alle MW-, WN- en NMP-gegevens plus eventuele VV-gegevens.)
Hint: gebruik de outer join, gedefinieerd in Opgave 2.93.

(f) Geef het totaal geprognosticeerde aantal behandelingen voor het lopende jaar.

(g) Geef het totaal aantal ligdagen in oktober 1987. (Het aantal *ligdagen* van een opname is het aantal dagen dat die opname heeft geduurd, inclusief de dag van opname en de dag van ontslag.)

(h) Geef van elke kinderloze patiënte geboren in en wonende te Dodewaard het beknopte medische dossier (zoals gedefinieerd in Opgave 4.12).
Hint: gebruik de in Opgave 4.12 ingevoerde uitdrukking BMD voor het beknopte medische dossier.

(i) Geef in een tupel over {OPNAMEN, BEHANDELINGEN, MEDICIJNEN} het totaal der factuurbedragen gesommeerd over alle Eindhovense patiëntes die in 1989 werden ingeschreven, uitgesplitst naar opnamen, patiëntbehandelingen en medicijnverstrekkingen.

(j) Geef het totaal van *alle* factuurbedragen over alle Eindhovense patiëntes die in 1989 werden ingeschreven.

(k) Geef in een tabel over {GESLACHT, BEHANDELINGEN} per geslacht de totale patiëntbehandelingskosten (= factuurbedragen) van alle Eindhovense patiënten die in 1989 werden ingeschreven.

6.2. Geef de volgende queries over UZKH in de vorm van een vraag of een opdracht weer.

(a) $\lambda v \in \text{UZKH} : \{t \in v(\text{MVS}) \mid t(\text{PNR}) = 256896\} \upharpoonright \{\text{MCD}, \text{DAT}\}$

(b) $(\lambda v \in \text{UZKH} : \text{'JA'})\ \theta$
$(\lambda v \in \text{UZKH} \text{ en } v(\text{AFD}) \upharpoonright \{\text{ANR}\} \subseteq v(\text{RU}) \upharpoonright \{\text{ANR}\} : \text{'NEE'})$

(c) $\lambda v \in \text{UZKH} : \{t \in v(\text{SP}) \mid \forall t' \in (\text{Hpbsp}(v) \text{ inv } t) :$
$\mid \{u \in (\text{Hpbsp}(v) \text{ inv } t) \mid u(\text{BCD}) = t'(\text{BCD})\} \mid \leq 5\}$

(d) $\lambda v \in \text{UZKH} : (\Sigma t \in v(\text{MED}) : (t(\text{VRD}) * t(\text{VKPR})) \text{ div } 100)$

6.3. Geef voor de vraag onder K3 van Voorbeeld 6.4 het analogon van oplossing (6) onder K1 in het geval dat (**) geldt.

6.4. Geef de vraag "Geef alle E-tupels waar volgens G geen E'-tupel met d'-waarde w' bijhoort" formeel weer. (Zie voor de context van de vraag de klasse K5 van Voorbeeld 6.4.)

6.5. Geef ook de vraag "Geef alle E-tupels waarbij volgens G in de E'-tabel geen enkele d'-waarde meer dan vijf keer voorkomt" formeel weer.

6.6. Geef de volgende vraag, die van dezelfde situatie uitgaat als Voorbeeld 6.4, formeel weer.
Geef alle E-tupels die volgens G in de E'-tabel ten minste alle d'-waarden hebben die het (eventuele) E-tupel t_0 ook heeft.

6.7. Deze opgave is een uitbreiding van K4 van Voorbeeld 6.6. We willen nu niet alleen een overzicht van alle tabelindexen met het bijbehorende aantal elementen maar daarnaast (in een tabel over {ENTITEIT, ATTRIBUUT, AANTAL}) ook een overzicht van alle attributen met het aantal verschillende waarden dat er voor dat attribuut in de betreffende tabel optreedt. Het totale antwoord dient daarbij te worden opgevat als een (tabelwaardige) functie over {ENTITEITEN, ATTRIBUTEN}.

6.8. Geef voor vraag V3 in Voorbeeld 6.5 een oplossing door in de daar gegeven oplossing (2) het cartesisch produkt te vervangen door een join en de DB-functie Hovropn weg te werken (met behulp van de definitie in §4.3).

6.9. (a) Geef voor $p \in \mathbb{N}$ en $d \in$ DATVZ een formele definitie van de functie VPV(p,d) over UZKH die aan elke $v \in$ UZKH toevoegt het in vraag V4 van Voorbeeld 6.5 omschreven *verpleegkundige verslag* van de (eventuele) opname op datum d van de patiënt met patiëntnummer p.

(b) Geef voor $p \in \mathbb{N}$ een formele definitie van de functie VPVS(p) over UZKH die aan elke $v \in$ UZKH een tabel over {OPNDAT, VERSLAG} toevoegt waarin de opnamedatum en het verpleegkundige verslag van elke opname van de (eventuele) patiënt p voorkomt. Maak hierbij gebruik van de functie VPV(p,d) uit onderdeel (a).

6.10. In ons ziekenhuis van Hoofdstuk 4 wordt onder *het dossier van een opname* verstaan een functie over {OPNGEG, OSPGEG, PBGEG} met als waarden achtereenvolgens het opnametupel in kwestie, de verzameling van alle overdrachtstupels behorende bij die opname en, ten slotte, de verzameling van alle behandelingstupels van de betreffende patiënt tijdens die opname (behandelingen op opname- of ontslagdatum moeten worden uitgesloten). Geef nu de volgende opdracht over UZKH formeel weer:

Geef alle opnamedossiers van patiënt 123453.

6.11. In ons ziekenhuis van Hoofdstuk 4 wil men voor elke maand van elk van de jaren 1960 tot en met 1988 het totaal der factuurbedragen weten - het totaal naar beneden afgerond op hele guldens - van de ontslagen die die maand plaatsvonden. Echter, sommigen willen als antwoord een tabel (met 30 attributen en 12 elementen) zoals gepresenteerd in Figuur 6.1(a), en sommigen willen als antwoord een tabel (met 13 attributen en 29 elementen) zoals gepresenteerd in Figuur 6.1(b).

MAAND	60	61	...	87	88
1 . . 12					

(a)

JAAR	1	2	...	11	12
60 . . 88					

(b)

Figuur 6.1

Geef elk van de beide wensen als een query over UZKH weer.

□

6.2 VIEWS

Wanneer een bepaalde query herhaaldelijk zal worden toegepast in een organisatie, dan kan het nuttig zijn die query van een naam te voorzien en in het vervolg die naam te gebruiken (in plaats van de query telkens opnieuw te specificeren). Zo'n "named query" wordt wel een *view* genoemd (omdat het zogezegd een "zicht op de database" is). Formeel beschouwen we een view als een geordend paar waarvan de tweede coördinaat een query is:

DEFINITIE 6.2:
Als U een verzameling is, dan:
p is een **view op** U $\overset{D}{\Longleftrightarrow}$ p is een geordend paar en
$\pi_2(p)$ is een query over U.

Als p een view is, zeg $p = (n\ ;\ q)$, dan noemen we n wel *de naam van p* en q wel *de definitie van p*.

Omdat in de praktijk verschillende queries niet een ad hoc karakter hebben maar regelmatig opnieuw toegepast worden, willen we gewoonlijk een hele *verzameling* queries "inblikken" in de vorm van views. Voorwaarde hierbij is wel dat de namen van de views onderling verschillend zijn. Anders gezegd, zo'n verzameling van views dient een functie te zijn. Zo'n functie die aan viewnamen de bijbehorende queries over U toevoegt, noemen we wel een *viewstelsel op U*:

DEFINITIE 6.3:
Als U een verzameling is, dan:
V is een **viewstelsel op** U $\overset{D}{\Longleftrightarrow}$ V is een functie en
$\forall p \in V : p$ is een view op U.

Een alternatieve verwoording is dat V een viewstelsel op U is desda V een functie is en $\forall n \in \mathrm{dom}(V) : V(n)$ is een query over U.

VOORBEELD 6.7:

We definiëren een viewstelsel VS1 op het DB-universum UZKH uit Hoofdstuk 4. Het viewstelsel bestaat uit zeven views. De viewnamen zijn characterstrings. De queries zijn voor een deel ontleend aan de voorbeelden in §6.1.

De views hebben de volgende namen en informele omschrijvingen:

‘VOORRAADOVERZICHT’ : code, naam en voorraad van alle medicijnen (zie Voorbeeld 6.2, K2(B)).

‘AFDELINGEN’ : de afdelingsgegevens inclusief het aantal werknemers per afdeling.

‘PATIENTKOSTEN’ : van elke Eindhovense patiënt die in 1989 werd ingeschreven, het patiëntnummer, de naam, de geboortedatum, het geslacht en het totaal der factuurbedragen, dit laatste uitgesplitst naar opnamen, behandelingen en medicijnverstrekkingen.

‘KWARTAALKOSTEN’ : het totale bedrag aan uit te betalen inkomensbestanddelen per kwartaal (op basis van de huidige bedragen).

‘PZVIEW’ : het voor de afdeling Personeelszaken relevante subschema (zie Voorbeeld 6.6, K2(B)).

‘DBDUMP’ : de gehele database-toestand (zie Voorbeeld 6.6, K1).

‘DBSIZE’ : overzicht van alle tabelindexen met het bijbehorende aantal elementen (zie Voorbeeld 6.6, K4).

De definitie van ons viewstelsel VS1 op UZKH luidt nu als volgt:

$$
\begin{array}{l}
VS1 = \{(\text{‘VOORRAADOVERZICHT’};\\
\qquad \lambda v \in \text{UZKH} : v(\text{MED}) \upharpoonright \{\text{MCD}, \text{MNM}, \text{VRD}\}),\\
\quad (\text{‘AFDELINGEN’};\\
\qquad \lambda v \in \text{UZKH} : \{\, a \cup \{(\text{WNAANTAL};\\
\qquad\qquad\qquad\qquad \mid \{m \in v(\text{MW}) \bowtie v(\text{WN}) \mid m(\text{ANR}) = a(\text{ANR})\} \mid)\}\\
\qquad\qquad\qquad \mid a \in v(\text{AFD})\}),\\
\quad (\text{‘PATIENTKOSTEN’};\\
\qquad \lambda v \in \text{UZKH} : \{\{(\text{NUMMER} \ ; p(\text{PNR})),\\
\qquad\qquad\qquad (\text{NAAM} \ ; p(\text{NAW})(\text{NM})),\\
\qquad\qquad\qquad (\text{GEBDAT} \ ; p(\text{GBD})),\\
\qquad\qquad\qquad (\text{GESLACHT}; p(\text{GESL})),\\
\qquad\qquad\qquad (\text{OPNBEDR} \ ; \Sigma t \in v(\text{ONT}) \text{ en } t(\text{PNR}) = p(\text{PNR}) : t(\text{BEDR})),\\
\qquad\qquad\qquad (\text{BEHBEDR} \ ; \Sigma t \in v(\text{PB}) \text{ en } t(\text{PNR}) = p(\text{PNR}) : t(\text{BEDR})),\\
\qquad\qquad\qquad (\text{MVSBEDR} \ ; \Sigma t \in v(\text{MVS}) \text{ en } t(\text{PNR}) = p(\text{PNR}) : t(\text{BEDR}))\}\\
\qquad\qquad \mid p \in v(\text{PAT}) \text{ en } p(\text{INSD}) \text{ div } 10000 = 89 \text{ en}\\
\qquad\qquad\ \ p(\text{NAW})(\text{WPL}) = \text{‘Eindhoven’}\}),
\end{array}
$$

$$
\begin{array}{l}
(\text{‘KWARTAALKOSTEN’};\\
\quad \lambda v \in \text{UZKH} : 13 * (\Sigma t \in v(\text{SP}) \text{ en } t(\text{IND}) = \text{‘JA’} : t(\text{HONU}) * t(\text{AUW}))\\
\qquad\qquad\qquad +3 * (\Sigma t \in v(\text{WN}) : t(\text{SAL})) * 100\\
\qquad\qquad\qquad +3 * (\Sigma t \in v(\text{NMP}) : t(\text{VERG})) * 100)\,,\\
(\text{‘PZVIEW’};\\
\quad \lambda v \in \text{UZKH} : v \upharpoonright \{\text{MW, WN, MP, NMP, VV}\} \cup\\
\qquad\qquad\qquad \{(\text{SP}; v(\text{SP}) \Upharpoonright \{\text{LOC}, \text{ABD}\})\})\,,\\
(\text{‘DBDUMP’};\\
\quad \text{id}(\text{UZKH}))\,,\\
(\text{‘DBSIZE’};\\
\quad \lambda v \in \text{UZKH} : \{\{(\text{ENTITEIT}; E)\,,\\
\qquad\qquad\qquad\quad (\text{AANTAL} \;;\, |\, v(E) \,|)\}\\
\qquad\qquad\qquad |\; E \in \text{dom}(\text{GZKH})\})\,\}.
\end{array}
$$

Nu het viewstelsel VS1 eenmaal gegeven is, kunnen we er vervolgens bij het formuleren van nieuwe queries gebruik van maken. Als voorbeeld bekijken we de opdracht "Geef nummer, naam en totaal der factuurbedragen van elke in 1989 ingeschreven patiënt uit Eindhoven aan wie al voor meer dan een 'vuurtoren' aan medicijnen is verstrekt". Met behulp van VS1 kunnen we de bijbehorende query eenvoudig als volgt specificeren:

$$
\begin{array}{l}
\lambda v \in \text{UZKH} : \{\, t \upharpoonright \{\text{NUMMER}, \text{NAAM}\} \cup\\
\qquad\qquad \{(\text{TOTAAL}; t(\text{OPNBEDR}) + t(\text{BEHBEDR}) + t(\text{MVSBEDR}))\}\\
\qquad\quad |\; t \in \text{VS1}(\text{‘PATIENTKOSTEN’})\,(v) \text{ en } t(\text{MVSBEDR}) > 25000\}
\end{array}
$$

Zonder de hulp van VS1 is de query duidelijk minder eenvoudig te formuleren:

$$
\begin{array}{l}
\lambda v \in \text{UZKH} : \{\;\{\;(\text{NUMMER}; p(\text{PNR}))\,,\\
\qquad\qquad\qquad (\text{NAAM} \quad ; p(\text{NAW})\,(\text{NM}))\,,\\
\qquad\qquad\qquad (\text{TOTAAL} \;\; ; \Sigma t \in v(\text{ONT}) \cup v(\text{PB}) \cup v(\text{MVS}) \text{ en}\\
\qquad\qquad\qquad\qquad\qquad t(\text{PNR}) = p(\text{PNR}) : t(\text{BEDR}))\}\\
\qquad\qquad |\; p \in v(\text{PAT}) \text{ en } p(\text{INSD}) \text{ div } 10000 = 89 \text{ en}\\
\qquad\qquad\; p(\text{NAW})\,(\text{WPL}) = \text{‘Eindhoven’ en}\\
\qquad\qquad (\Sigma t \in v(\text{MVS}) \text{ en } t(\text{PNR}) = p(\text{PNR}) : t(\text{BEDR})) > 25000\}
\end{array}
$$

□ Voorbeeld 6.7.

We merken op dat de views genaamd 'DBDUMP' en 'DBSIZE' in VS1 van zo'n algemeen karakter zijn dat ze bij een DBMS in feite standaard geleverd zouden moeten worden.

Een view (n ; q) kan ook worden opgevat als een "kennisregel" waarbij n dan het te definiëren "begrip" is en q de definitie van dat begrip. Een duidelijk voorbeeld hiervan is de (voor de salarisadministratie bestemde) view genaamd 'KWARTAALKOSTEN' in VS1, waarin nauwkeurig de gedetailleerde kennis is vastgelegd welke ingrediënten (in welke verhoudingen) de totale salariskosten per kwartaal bepalen. Met behulp van views kan zodoende een hele "bibliotheek" van afleidbare begrippen worden opgebouwd rondom een database-universum. Omdat een database-managementsysteem per gedefinieerd database-universum vaak meer dan één viewstelsel toestaat, bijvoorbeeld één viewstelsel per gebruiker of gebruikersgroep, kunnen er meestal zelfs verschillende soorten bibliotheken worden aangelegd.

Met ons ruime viewbegrip modelleren we niet alleen de *SQL-views* ([SQL 89]), waarvan de view genaamd 'VOORRAADOVERZICHT' in VS1 een typisch voorbeeld is, maar modelleren we tevens de ANSI/SPARC *external views* ([TK 78]), getuige bijvoorbeeld de view genaamd 'PZVIEW' in VS1.

Het kan in de praktijk nuttig zijn om bij het definiëren van nieuwe views gebruik te maken van reeds eerder gedefinieerde views (zoals ook in de opgaven zal blijken). De meeste database-managementsystemen bieden deze mogelijkheid dan ook, zie bijvoorbeeld Kijkgat 6 in [IDT 88].

OPGAVEN

6.12. Geef de volgende opdrachten over het DB-universum UZKH uit Hoofdstuk 4 formeel weer, voorzie ze van een naam en leg ze vast in een viewstelsel VS2 op UZKH. Hoewel sommige van deze opdrachten reeds in Opgave 6.1 voorkwamen, wordt u geacht nu gebruik te maken van het viewstelsel VS1 gedefinieerd in Voorbeeld 6.7.

(a) Geef in een tupel over {OPNAMEN, BEHANDELINGEN, MEDICIJNEN} het totaal der factuurbedragen gesommeerd over alle Eindhovense patiëntes die in 1989 werden ingeschreven, uitgesplitst naar opnamen, patiëntbehandelingen en medicijnverstrekkingen.

(b) Geef het totaal van *alle* factuurbedragen over alle Eindhovense patiëntes die in 1989 werden ingeschreven.

(c) Geef in een tabel over {GESLACHT, BEHANDELINGEN} per geslacht de totale patiëntbehandelingskosten van alle Eindhovense patiënten die in 1989 werden ingeschreven.

(d) Geef de gemiddelde *kwartaalkosten* per (in dienst zijnde) medewerker, naar beneden afgerond op hele guldens.

6.13. Leg de volgende opdrachten over het DB-universum UZKH uit Hoofdstuk 4 (met zelfgekozen namen) vast als views in een viewstelsel VS3 op UZKH. Gebruik hierbij, voor zover mogelijk, de viewstelsels VS1 uit Voorbeeld 6.7 en VS2 uit de vorige opgave.

(a) Geef in een tabel over {GESLACHT, BEHANDELINGEN} voor de twee geslachten de onderlinge percentages van de totale patiëntbehandelingskosten van alle Eindhovense patiënten die in 1989 werden ingeschreven (naar beneden afgerond op tienden van procenten).

(b) Geef van elk medisch personeelslid het identiteitsnummer, de naam, het salaris en het verschil ten opzichte van de gemiddelde kwartaalkosten per in dienst zijnde medewerker (positief als het salaris meer dan het gemiddelde is en negatief als het minder is).

(c) Geef in een tabel over {1, 2} alle patiëntbehandelingsparen waarvan de eerste patiëntbehandeling een directe aangever van eventuele besmettingen voor de tweede is (in de zin van Voorbeeld 6.5, V5).
Hint: zie ook Opgave 1.13.

6.14. Leg de volgende opdrachten over UZKH, die zijn gebaseerd op vraag V5 van Voorbeeld 6.5, met zelfgekozen namen vast als (recursieve) views in een viewstelsel VS4 op UZKH. U mag hierbij gebruik maken van de eerder gedefinieerde viewstelsels.

(a) Geef de patiëntgegevens van alle patiënten die op directe of indirecte wijze via een behandeling besmet kunnen zijn geraakt door specialist 109.

(b) Geef de identiteitsnummers van alle specialisten die sinds 3 juli 1989 op directe of indirecte wijze via een behandeling besmet kunnen zijn geraakt als gevolg van een besmettingshaard in behandelruimte 'XP4'.

□

7 ONDERHOUD

7.1 TRANSACTIES

De toestand van een organisatie zal gewoonlijk nogal aan verandering onderhevig zijn: medewerkers, klanten en produkten komen en gaan, bestellingen en facturen komen binnen en worden verwerkt, en salarissen, voorraden en prijzen stijgen of dalen en passant. De administratieve weerslag van zulke veranderingen - in database-kringen wel *onderhoud* genoemd - kunnen we formeel beschrijven met behulp van functies die aan elke mogelijke DB-toestand die DB-toestand toevoegen die de nieuwe situatie zou reflecteren. We zullen zulke functies *transacties* noemen. Onze (zeer) algemene definitie luidt als volgt:

DEFINITIE 7.1:
Als U een verzameling is, dan:
f is een **transactie op** U $\overset{D}{\Longleftrightarrow} f \in U \rightarrow U$.

VOORBEELD 7.1:

We lichten het transactiebegrip toe aan de hand van het DB-universum VBU uit Voorbeeld 1.3 en het volgende tupel:

t4 = {(ANR; 3), (NAAM; 'INKOOP'), (MANNR; 7)}

Uitgaande van VBU zouden we (in onze naïviteit) het toevoegen van het afdelingstupel t4 in eerste instantie misschien met behulp van de volgende functie f5 willen beschrijven:

$$f5 = \lambda v \in \text{VBU}: \{(\text{MEDEW}; v(\text{MEDEW})),\ (\text{AFD}\ ; v(\text{AFD}) \cup \{t4\})\}.$$

De functie f5 blijkt echter geen transactie op VBU te zijn! Weliswaar is f5 een functie *over* VBU maar niet *naar* VBU: Voor sommige v in VBU is $f5(v)$ namelijk geen element van VBU. Dit doet zich in het bijzonder voor als $t4 \notin v(\text{AFD})$ terwijl het afdelingsnummer 3 of de afdelingsnaam 'INKOOP' wel in v(AFD) voorkomt of het medewerkersnummer 7 niet in v(MEDEW) voorkomt. (Zie in dit verband de eisen (2), (3) en (5) uit Voorbeeld 1.3.) Als het in de genoemde speciale gevallen wenselijk is om de toestand te laten zoals het was, dan zou de beschrijving van de aldus genuanceerde toevoegings*poging* er als volgt uit kunnen zien:

$$f6 = \lambda v \in \text{VBU}: \begin{cases} f5(v) & \text{als } f5(v) \in \text{VBU} \\ v & \text{in het andere geval} \end{cases}$$

De functie f6 is wél een transactie op VBU.

Als er ook nog dynamische constraints op VBU zouden zijn vastgelegd, dan moest bovendien gelden dat $(v; f5(v))$ een toegestane transitie is. Stel bijvoorbeeld dat er alleen in bepaalde situaties nieuwe afdelingen bij mogen komen, bijvoorbeeld alleen als het aantal medewerkers een tienvoud of een tienvoud plus één bedraagt. (De werkelijkheid gaat wat dat betreft vaak onze fantasie te boven.) De gewenste transitierelatie is dan

$$\begin{aligned} R_0 = \{(v\,; v') \mid\ & (v\,; v') \in \text{VBU} \times \text{VBU en} \\ & \text{als } |v(\text{MEDEW})| \bmod 10 \notin \{0,1\} \\ & \text{dan id}(\{\text{ANR}\}) \text{ verbindt } v'(\text{AFD}) \text{ met } v(\text{AFD})\}. \end{aligned}$$

(Terzijde merken we op dat dit tevens een voorbeeld van een conditionele verbinding is.)

Als we in het geval van een niet toegestane transitie de toestand weer willen laten zoals het was, dan is de bedoelde transactie dus:

$$f7 = \lambda v \in \text{VBU}: \begin{cases} f5(v) & \text{als } (v\,; f5(v)) \in R_0 \\ v & \text{in het andere geval} \end{cases}$$

We merken op dat $(v\,; f5(v)) \in f5$, zodat we f7 met behulp van Definitie 0.16 ook als volgt kunnen herschrijven:

$$\begin{aligned} f7 &= \{(v\,; f5(v)) \mid v \in \text{VBU en } (v\,; f5(v)) \in R_0\} \cup \\ &\quad \{(v\,; v) \quad\;\; \mid v \in \text{VBU en } (v\,; f5(v)) \notin R_0\} \\ &= \text{id(VBU)}\ \theta\ (R_0 \cap f5). \end{aligned}$$

□ Voorbeeld 7.1.

In Voorbeeld 7.1 brachten we naar voren dat een transactie ook aan de eventuele dynamische constraints moet voldoen en daarom moet "passen" binnen de betreffende transitierelatie R_0. We spreken dan wel van een *transactie binnen* R_0. Formeel:

DEFINITIE 7.2:
Als R een relatie is, dan:
f is een **transactie binnen** $R \overset{D}{\Leftrightarrow} f$ is een functie en $f \subseteq R$.

Als we Definitie 7.1 en Definitie 7.2 met elkaar vergelijken dan zien we dat elke transactie *op U* tevens een transactie *binnen* $U \times U$ is. Het omgekeerde hoeft niet het geval te zijn omdat het domein van een transactie binnen $U \times U$ niet noodzakelijk geheel U is (of, anders gezegd, omdat de transactie niet noodzakelijk in elke toestand gedefinieerd is).

In Voorbeeld 7.1 is

- f5 geen transactie op VBU,
- f6 wel een transactie op VBU maar geen transactie binnen R_0, en
- f7 wel een transactie binnen R_0.

De aanpassingen die we in Voorbeeld 7.1 van f5 maakten om te komen tot een transactie f7 binnen de transitierelatie R_0 zijn van een meer algemeen belang. We voeren daarom het volgende begrip in:

DEFINITIE 7.3:
Als U een verzameling is en $R \subseteq U \times U$ en f is een functie, dan:
$\text{Ad}(U, R, f) \overset{D}{=} \text{id}(U)\ \theta\ (R \cap f)$.

We noemen $\text{Ad}(U, R, f)$ wel *de door U en R bepaalde adaptie van f*. We spreken van *de door U bepaalde adaptie van f* als er geen dynamische constraints zijn en we dus in feite te maken hebben met de transitierelatie $U \times U$, de meest "liberale" transitierelatie op U. We schrijven deze (uitsluitend door de statische constraints bepaalde) adaptie als $\text{Adap}(U, f)$:

DEFINITIE 7.4:
Als U een verzameling is en f is een functie, dan:
$\mathrm{Adap}(U, f) \stackrel{D}{=} \mathrm{Ad}(U, U \times U, f)$.

We geven de volgende alternatieve beschrijvingen van de twee zojuist ingevoerde begrippen:

$$\mathrm{Adap}(U, f) = \lambda v \in U : \begin{cases} f(v) & \text{als } v \in \mathrm{dom}(f) \text{ en } f(v) \in U \\ v & \text{in het andere geval} \end{cases}$$

$$\mathrm{Ad}(U, R, f) = \lambda v \in U : \begin{cases} f(v) & \text{als } v \in \mathrm{dom}(f) \text{ en } (v\,;f(v)) \in R \\ v & \text{in het andere geval} \end{cases}$$

Als "het andere geval" zich voordoet en de veranderingspoging dus als het ware "teruggefloten" wordt, dan spreekt men wel van een *rollback* van de veranderingspoging.

Terugblikkend op Voorbeeld 7.1 zien we dat

$$\begin{aligned} &\mathrm{Adap}(\mathrm{VBU}, \mathrm{f5}) = \mathrm{f6} \text{ en} \\ &\mathrm{Ad}(\mathrm{VBU}, \mathrm{R_0}, \mathrm{f5}) = \mathrm{f7}. \end{aligned}$$

STELLING 7.1:
Als U een verzameling is en $R \subseteq U \times U$ en f is een functie, dan:

(a) $\mathrm{Adap}(U, f)$ is een transactie op U;

(b) $\mathrm{Ad}(U, R, f)$ is een transactie op U;

(c) als R reflexief op U is, dan is $\mathrm{Ad}(U, R, f)$ een transactie binnen R;

(d) als f een transactie op U en binnen R is, dan $\mathrm{Ad}(U, R, f) = f$;

(e) als R reflexief op U is, dan $\mathrm{Ad}(U, R, \mathrm{Ad}(U, R, f)) = \mathrm{Ad}(U, R, f)$.

Bewijs:
Omdat f een functie is, is $R \cap f$ dat volgens Lemma 0.3(a) ook.
Verder is gegeven dat $R \subseteq U \times U$, dus ook $R \cap f \subseteq U \times U$.
Uit Definitie 7.3 en Lemma 0.12(a) volgt nu dat $\mathrm{Ad}(U, R, f)$ een functie over U is, en wel een element van $U \twoheadrightarrow U$.
Hiermee is zowel (b) als het meer speciale geval (a) bewezen.

Als R reflexief op U is, dan $\mathrm{id}(U) \subseteq R$;
uit Definitie 7.3 volgt dan dat $\mathrm{Ad}(U, R, f) \subseteq R$.
Hiermee is (c) bewezen.

f is een transactie binnen R, dus $f \subseteq R$. (1)
f is een transactie op U, dus $\mathrm{dom}(f) = U$. (2)
Toepassing van Definitie 7.3, van (1) en van (2) hierboven levert nu achtereenvolgens:

$$\begin{aligned}\mathrm{Ad}(U, R, f) &= \mathrm{id}(U)\ \theta\ (R \cap f)\\ &= \mathrm{id}(U)\ \theta\ f\\ &= f.\end{aligned}$$

Hiermee is (d) bewezen.

Onderdeel (e) volgt direct uit de onderdelen (b), (c) en (d).

□

We hebben de voorgaande adaptiebegrippen ingevoerd mede omdat ze goed modelleren hoe de interactie tussen een gebruiker en een database management systeem er in het algemeen uit zou kunnen zien: bij elke functie f die door een gebruiker als onderhoudsopdracht (dus als een speciaal soort "applicatie") wordt opgegeven, zou het DBMS de facto uitvoering kunnen geven aan de adaptie van f bepaald door het DB-universum en de transitierelatie zoals die aan het DBMS bekend zijn. Dit betekent dus dat elke statische en dynamische constraint die in het DBMS in kwestie specificeerbaar is niet in al die afzonderlijke applicaties hoeft te worden "vervlochten" maar centraal in het DBMS kan worden afgevangen. Voor de mate waarin we allerlei verfijnde constraints kunnen opgeven aan bestaande DBMS-en bij het specificeren van DB-universa en transitierelaties verwijzen we naar Kijkgat 4 in [NGI 88] en [IDT 88].

We willen er heel nadrukkelijk op wijzen dat het achter elkaar uitvoeren van de adapties van twee functies f en g zeker niet hetzelfde resultaat hoeft op te leveren als het uitvoeren van de adaptie van de samengestelde functie $g \circ f$ als één "ondeelbaar geheel" (alias één *unit of work*). Anders gezegd, $\mathrm{Ad}(U, R, g) \circ \mathrm{Ad}(U, R, f)$ hoeft niet dezelfde functie te zijn als de transactie $\mathrm{Ad}(U, R, g \circ f)$. Om dit duidelijk te maken werken we beide functies uit en illustreren we een en ander bovendien met een voorbeeld. We zullen hierbij zelfs aannemen dat $U \subseteq \mathrm{dom}(f) \cap \mathrm{dom}(g)$ en $\mathrm{rng}(f) \subseteq \mathrm{dom}(g)$, dus dat f en g voor elke DB-toestand van U gedefinieerd zijn en dat g bovendien gedefinieerd is voor elk resultaat van f, ook als dat geen DB-toestand conform U is.

$$\mathrm{Ad}(U, R, g \circ f) = \lambda v \in U : \begin{cases} g(f(v)) & \text{als } (v\,; g(f(v))) \in R & \text{(A)} \\ v & \text{als } (v\,; g(f(v))) \notin R & \text{(B)} \end{cases}$$

$$\mathrm{Ad}(U, R, g) \circ \mathrm{Ad}(U, R, f)$$

$$= \lambda v \in U : \begin{cases} \mathrm{Ad}(U, R, g)\,(f(v)) & \text{als } (v\,; f(v)) \in R \\ \mathrm{Ad}(U, R, g)\,(v) & \text{als } (v\,; f(v)) \notin R \end{cases}$$

$$= \lambda v \in U : \begin{cases} g(f(v)) & \text{als } (v\,; f(v)) \in R \text{ en } (f(v)\,; g(f(v))) \in R & (++) \\ f(v) & \text{als } (v\,; f(v)) \in R \text{ en } (f(v)\,; g(f(v))) \notin R & (+-) \\ g(v) & \text{als } (v\,; f(v)) \notin R \text{ en } (v\,; g(v)) \in R & (-+) \\ v & \text{als } (v\,; f(v)) \notin R \text{ en } (v\,; g(v)) \notin R & (--) \end{cases}$$

In principe kunnen alle acht combinaties (A++, A+−, A−+, A−−, B++, B+−, B−+ en B−−) zich voordoen. Echter, alleen in de situaties A++ en B−− leveren beide functies met zekerheid dezelfde resultaten op.

Ook als er geen dynamische constraints zijn, treden er de nodige verschillen op:

$$\mathrm{Adap}(U, g \circ f) = \lambda v \in U : \begin{cases} g(f(v)) & \text{als } g(f(v)) \in U & \text{(A)} \\ v & \text{als } g(f(v)) \notin U & \text{(B)} \end{cases}$$

$$\mathrm{Adap}(U, g) \circ \mathrm{Adap}(U, f) = \lambda v \in U : \begin{cases} g(f(v)) & \text{als } f(v) \in U \text{ en } g(f(v)) \in U & (++) \\ f(v) & \text{als } f(v) \in U \text{ en } g(f(v)) \notin U & (+-) \\ g(v) & \text{als } f(v) \notin U \text{ en } g(v) \in U & (-+) \\ v & \text{als } f(v) \notin U \text{ en } g(v) \notin U & (--) \end{cases}$$

Op A+− en B++ na kunnen verder alle zes andere situaties zich voordoen. Ook hier geldt weer dat alleen in de situaties A++ en B−− beide functies met zekerheid dezelfde resultaten opleveren.

VOORBEELD 7.2:

We definiëren twee functies f72 en g72 en schetsen met behulp van Figuur 7.1

(a) hoe *zonder* dynamische constraints elk van de situaties A++, A−+, A−−, B+−, B−+ en B−− zich kan voordoen, en

(b) hoe *met* dynamische constraints de onder (a) geschetste situaties mogelijk kunnen blijven en daarnaast ook de situaties A+− en B++ kunnen optreden.

We nemen het DB-universum VBU uit Voorbeeld 1.3 weer als uitgangspunt en bij (b) gebruiken we de transitierelatie R_0 gedefinieerd in Voorbeeld 7.1.

De functie f72 beschrijft het toevoegen van een medewerkster Vos bij afdeling 2 met een salaris van *f* 3300. Voor het medewerkersnummer wordt hierbij het aantal medewerkers plus één gekozen (dit in de stilzwijgende, doch niet noodzakelijk juiste, veronderstelling dat de medewerkers opeenvolgend zijn genummerd vanaf 1).

De functie g72 beschrijft het toevoegen van de afdeling Marketing met medewerker 2 als manager. Voor het afdelingsnummer wordt hierbij het aantal afdelingen plus één gekozen (dit in de veronderstelling dat ook afdelingen opeenvolgend zijn genummerd vanaf 1).

Merk op dat de toe te voegen tupels in feite deels afhankelijk zijn van de toestand waaraan ze worden toegevoegd. Die afhankelijkheid wordt beschreven door de hulpfuncties f71 en g71 die we eerst zullen invoeren.

Het domein van de functies g71 en g72 willen we zo ruim kiezen dat zowel VBU als rng(f72) er binnen vallen, dit om onze aannamen (vlak voor de introductie van de gevalsonderscheidingen) gestand te doen. We kiezen daartoe een DB-universum VBU7 zonder databaseconstraints en zonder tupel- en tabelconstraints voor de tabelindex MEDEW.

De definities luiden nu als volgt:

$$\begin{aligned} VBU7 = \Pi(\{&(MEDEW;\ \mathbf{P}(\Pi(FM))),\\ &(AFD\ ;\ WA)\}); \end{aligned}$$

$$\begin{aligned} f71 = \lambda v \in VBU\ \ &: \{(NR;\ |v(MEDEW)| + 1),\ (AFDNR;\ 2),\\ &\quad (NAAM;\ \text{'VOS'}),\ (SAL;\ 3300),\ (GESL;\ \text{'V'})\};\\ g71 = \lambda v \in VBU7 &: \{(ANR;\ |v(AFD)| + 1),\ (MANNR;\ 2),\\ &\quad (NAAM;\ \text{'MARKETING'})\}; \end{aligned}$$

$$\begin{aligned} f72 &= \lambda v \in \mathrm{VBU} &&: v\ \theta\ \{(\mathrm{MEDEW};\ v(\mathrm{MEDEW}) \cup \{f71(v)\})\}; \\ g72 &= \lambda v \in \mathrm{VBU7} &&: v\ \theta\ \{(\mathrm{AFD}\ ;\ v(\mathrm{AFD}) \cup \{g71(v)\})\}. \end{aligned}$$

In Figuur 7.1 vermelden we voor elk van de mogelijke situaties aan welke voorwaarden wel ("+") en aan welke niet ("–") voldaan moet zijn door een DB-toestand om deze situatie te laten optreden. De tussen haakjes geplaatste plussen en minnen zijn hierbij geen *extra* eisen maar *consequenties* van de andere eisen. Naast de in Figuur 7.1 aangegeven eisen stellen we aan elk van die DB-toestanden nog de eis dat daarin nog geen afdeling Marketing voorkwam ('MARKETING' $\notin \{t(\mathrm{NAAM}) \mid t \in v(\mathrm{AFD})\}$).

Elk van de acht situaties werd gekarakteriseerd door het al dan niet toe kunnen passen van de functies f72, g72 en g72 $\circ$ f72. De voorwaarden hiervoor hebben we in Figuur 7.1 gemakshalve herhaald.

We maken in Figuur 7.1 gebruik van de volgende afkortingen:

$$\begin{aligned} m_v &= |\, v(\mathrm{MEDEW}) \,| + 1 \\ a_v &= |\, v(\mathrm{AFD}) \,| + 1 \\ M_v &= \{t(\mathrm{NR}) \mid t \in v(\mathrm{MEDEW})\} \\ A_v &= \{t(\mathrm{ANR}) \mid t \in v(\mathrm{AFD})\} \end{aligned}$$

	A++	A−+	A−−	B+−	B−+	B−−
g72(f72(v)) $\in$ VBU	+	+	+	–	–	–
f72(v) $\in$ VBU	+	–	–	+	–	–
g72(v) $\in$ VBU		+	–		+	–
$m_v \in M_v$?	–	–	–	–	+	+
$2 \in M_v$?	(+)	+	–		+	
$2 \in M_v \cup \{m_v\}$?	+	(+)	+		(+)	
$a_v \in A_v$?	–	–	–	+	–	+
$2 \in A_v$?	+	–	–	+		
$2 \in A_v \cup \{a_v\}$?	(+)	+	+	(+)		

(a) Mogelijke situaties *zonder* dynamische constraints

	A++	A−+	A−−	B+−	B−+	B−−	A+−	B++
$(v;\ g72(f72(v))) \in R_0$	+	+	+	−	−	−	+	−
$(v;\ f72(v)) \in R_0$	+	−	−	+	−	−	+	+
$(v;\ g72(v)) \in R_0$		+	−		+	−		
$(f72(v);\ g72(f72(v))) \in R_0$	+			−			−	+
(a) $m_v \in M_v$?	−	−	−	−	+	+	−	−
$2 \in M_v$?	(+)	+	−		+			(+)
$2 \in M_v \cup \{m_v\}$?	+	(+)	+		(+)		+	+
$a_v \in A_v$?	−	−	−	+	−	+	−	−
$2 \in A_v$?	+	−	−	+			+	+
$2 \in A_v \cup \{a_v\}$?	(+)	+	+	(+)			(+)	(+)
(b) $\mid v(\text{MEDEW}) \mid$ mod 10?	0	0 of 1	0 of 1	0 of 1	0 of 1	0 of 1	1	9
$m_v = 2$?	(−)	(−)	(+)					(−)
$a_v = 2$?	(−)	(+)	(+)				(−)	(−)

(b) Mogelijke situaties *met* dynamische constraints

Figuur 7.1: Voldoende voorwaarden voor karakteristieke voorbeelden

Voor elk van de acht situaties zullen we nu een DB-toestand in VBU aangeven die aan alle voor het optreden van deze situatie noodzakelijke voorwaarden voldoet. We maken daartoe gebruik van Figuur 7.2. Deze figuur representeert een AFD-tabel met 3 tupels en een MEDEW-tabel met 11 tupels, met dien verstande dat de AFD-tabel nog de "parameters" α en β en de MEDEW-tabel nog de "parameters" γ en δ bevat.

	ANR	NAAM	MANNR
	1	PRODUCTIE	1
A−+, A−−			
	α	VERKOOP	4
	β	RESEARCH	5

(a) Een sjabloon voor enige AFD-tabellen

	NR	AFDNR	NAAM	GESL	SAL
	1	1	W.F. Hermans	M	3000
A−−					
	4	1	T. Hermans	M	3000
	5	1	M. Minco	V	3000
	6	1	K. van Kooten	M	3000
	7	1	M. van Keulen	V	3000
	8	1	M. Biesheuvel	M	3000
	9	1	H. Mulisch	M	3000
	γ	1	A.M.G. Schmidt	V	3000
	δ	1	G. Reve	M	3000
B++					
	14	1	R. Rubinstein	V	3000
A++					
	15	1	G. Krol	M	3000

(b) Een sjabloon voor enige MEDEW-tabellen

Figuur 7.2: Een sjabloon voor enige DB-toestanden in VBU

Met deze "tabel-sjablonen" als basis beschrijven we in Figuur 7.3 een achttal DB-toestanden in VBU door in de eerste plaats de aantallen AFD-tupels en MEDEW-tupels geschikt te kiezen en vervolgens de waarden voor de parameters α, β, γ en δ (voor zover van toepassing) geschikt te kiezen. In die situaties waarin we voor een DB-toestand minder dan drie afdelingen of minder dan elf medewerkers nodig hebben laten we "de tupels onderin" weg (zie de open ruimtes tussen de tupels in Figuur 7.2). Dus, om de situaties B+−, B−+, B−− en A+− te verkrijgen zullen we telkens alle drie AFD-tupels en alle elf MEDEW-tupels kiezen, voor de situatie A++ laten we medewerker 15 weg, voor B++ laten we de twee medewerkers 14 en 15 weg, voor A−+ volstaan we met afdeling 1 (en elf medewerkers) en voor A−− nemen we alleen afdeling 1 en medewerker 1.

		A++	A−+	A−−	B+−	B−+	B−−	A+−	B++
g72 ∘ f72 toepasbaar?		+	+	+	−	−	−	+	−
f72 toepasbaar?		+	−	−	+	−	−	+	+
g72 toepasbaar?		+	+	−	−	+	−	−	+
$\|v(\mathrm{AFD})\|$		3	1	1	3	3	3	3	3
$\|v(\mathrm{MEDEW})\|$		10	11	1	11	11	11	11	9
AFD:	α	2	nvt	nvt	2	2	3	2	2
	β	3	nvt	nvt	4	3	4	3	3
MEDEW:	γ	2	2	3	3	2	3	2	2
	δ	13	13	13	13	12	12	13	13

Figuur 7.3: Acht karakteristieke DB-toestanden in VBU, gebaseerd op Figuur 7.2

□ Voorbeeld 7.2.

OPGAVEN

7.1. In deze opgave refereren we aan de in Voorbeeld 7.1 ingevoerde functies f5 en f6. We definiëren f8 = Ad(VBU, VBR, f5), waarbij VBR de in Voorbeeld 5.1 gedefinieerde transitierelatie is.
Bewijs dat f8 = f6.

7.2. Als U een verzameling is en $R \subseteq U \times U$ en f is een functie, dan is Ad(U, R, f) een transactie binnen $R \cup \mathrm{id}(U)$. Bewijs dit.

7.3. Geef voor elk van de 12 "minnen" in Figuur 7.3 een reden aan.

□

7.2 ENIGE GANGBARE KLASSEN VAN TRANSACTIES

Om een indruk te geven van de gangbare transactiemogelijkheden van bestaande DBMS-en, en van de SQL2-standaard bijvoorbeeld (zie [SQL 89]), zullen we enige speciale klassen van transacties invoeren die tezamen het grootste deel van die bestaande mogelijkheden overdekken en in sommige opzichten zelfs aanzienlijk verder gaan. Het leeuwedeel van de huidige transactiemogelijkheden kenmerkt zich daardoor dat de veranderingen zich plegen te beperken tot slechts één enkele tabelindex tegelijk. Iets nauwkeuriger gezegd: De ter beschikking gestelde transacties zijn allemaal van de vorm $\mathrm{Ad}(U, R, f)$ waarbij f een functie over het DB-universum U is zodanig dat voor één bepaalde tabelindex E_0 van U het beeld $f(v)$ voor elke $v \in U$ een functie over de tabelindexenverzameling $\mathrm{He}(U)$ is met de eigenschap dat $f(v) \not\upharpoonright \{E_0\} = v \not\upharpoonright \{E_0\}$. Dit betekent dus dat we in deze gevallen alleen de veranderingen voor de E_0-tabel hoeven te specificeren. We voeren daarom het volgende hulpbegrip in:

DEFINITIE 7.5:
Als U een DB-universum is en $R \subseteq U \times U$ en $E \in \mathrm{He}(U)$ en τ is een functie over U, dan:
$\mathrm{Main}(U, R, E, \tau) \stackrel{D}{=} \mathrm{Ad}(U, R, \lambda v \in U : v \,\theta\, \{(E\,;\tau(v))\})$.

Dus $\mathrm{Main}(U, R, E, \tau)$ is de functie over U die aan elke DB-toestand v van U de toestand toevoegt waarin de E-tabel de waarde $\tau(v)$ heeft en de andere tabelindexen nog hun "oude" waarde hebben, althans als dit een toegestane transitie vormt; als deze transitie niet is toegestaan dan wordt aan v gewoon v zelf toegevoegd. We noemen $\mathrm{Main}(U, R, E, \tau)$ wel *de maintenance van E volgens* τ (binnen de grenzen van R op U). Ter illustratie merken we op dat de functie f7 uit Voorbeeld 7.1 te schrijven is als Main(VBU, R_0, AFD, $\lambda v \in \mathrm{VBU} : v(\mathrm{AFD}) \cup \{\mathrm{t4}\}$).

Met behulp van de speciale klasse van "ééntabel-transacties" die we zojuist in Definitie 7.5 hebben ingevoerd kunnen we nu op eenvoudige wijze de gangbare SQL-achtige transacties definiëren.

We beginnen met een elementaire "insert-operatie", het toevoegen van één gegeven tupel t aan de E-tabel (mits daarbij aan alle statische en dynamische constraints is voldaan):

DEFINITIE 7.6:
Als U een DB-universum is en $R \subseteq U \times U$ en $E \in \text{He}(U)$ en t is een functie, dan:
$\text{Inst}(U, R, E, t) \overset{D}{=} \text{Main}(U, R, E, \lambda v \in U : v(E) \cup \{t\})$.

Ter illustratie merken we op dat de functie f7 uit Voorbeeld 7.1 van deze vorm is en te schrijven is als Inst(VBU, R_0, AFD, t4). Hiermee is dus op compacte én correcte wijze het (eventueel) toevoegen van het tupel t4 aan de AFD-tabel binnen de grenzen van de transitierelatie R_0 op het DB-universum VBU gespecificeerd.

We vervolgen met de tweede gangbare soort insert-operatie, het toevoegen van een queryresultaat aan de E-tabel (mits daarbij weer aan alle statische en dynamische constraints is voldaan):

DEFINITIE 7.7:
Als U een DB-universum is en $R \subseteq U \times U$ en $E \in \text{He}(U)$ en q is een verzamelingsfunctie over U, dan:
$\text{Insq}(U, R, E, q) \overset{D}{=} \text{Main}(U, R, E, \lambda v \in U : v(E) \cup q(v))$.

Het is eenvoudig in te zien dat Definitie 7.6 een speciaal geval is van Definitie 7.7:
$\text{Inst}(U, R, E, t) = \text{Insq}(U, R, E, \lambda v \in U : \{t\})$.

De transactie Insq(U, R, E, q) specificeert dus de poging tot het "in één klap" toevoegen van de verzameling $q(v)$ aan de E-tabel en *niet* het één voor één pogen toe te voegen van de afzonderlijke elementen van de verzameling $q(v)$; het is dus een "alles of niets"-operatie.

VOORBEELD 7.3:

De poging tot het in één keer toevoegen van de tabel T1 uit Voorbeeld 1.1 (bevattende drie "kandidaat-medewerkers") aan de MEDEW-tabel binnen de in Voorbeeld 7.1 ingevoerde transitierelatie R_0 op het DB-universum VBU kunnen we specificeren met gebruikmaking van de volgende (constante) functie:

$\text{q70} = \lambda v \in \text{VBU: T1}$

De bedoelde transactie is nu Insq(VBU, R_0, MEDEW, q70). In geval van een DB-toestand $v \in$ VBU waarbij bijvoorbeeld één van de medewerkersnummers uit T1 al voorkomt in v(MEDEW) of één van de afdelingsnummers uit T1 niet voorkomt in v(AFD) zal dus geen enkel element van T1 worden toegevoegd.

Bij het één voor één pogen toe te voegen van de afzonderlijke elementen van T1 kan

het zijn dat sommige elementen wel en sommige elementen niet toegevoegd worden, bijvoorbeeld als afdeling 1 wel in v(AFD) voorkomt maar afdeling 2 niet. Als we de elementen van T1 even t7, t8 en t9 noemen, dan kunnen we de "batch" van pogingen om achtereenvolgens t7, t8 en t9 toe te voegen specificeren als

$$\text{Inst}(\text{VBU}, \text{R}_0, \text{MEDEW}, \text{t9}) \circ \text{Inst}(\text{VBU}, \text{R}_0, \text{MEDEW}, \text{t8}) \circ \text{Inst}(\text{VBU}, \text{R}_0, \text{MEDEW}, \text{t7})$$

□ Voorbeeld 7.3.

De volgende operatie betreft het verwijderen van een query-resultaat uit de E-tabel (mits ook daarbij weer aan alle constraints is voldaan). Zo'n operatie wordt wel een "delete-operatie" genoemd.

DEFINITIE 7.8:

Als U een DB-universum is en $R \subseteq U \times U$ en $E \in \text{He}(U)$ en q is een verzamelingsfunctie over U, dan:

$\text{Del}(U, R, E, q) \stackrel{D}{=} \text{Main}(U, R, E, \lambda v \in U : v(E) - q(v))$.

Het "leegmaken" van de E-tabel is hiervan een belangrijk speciaal geval: neem $q = \lambda v \in U : v(E)$.

VOORBEELD 7.4:

We geven enkele voorbeelden van delete-operaties aan de hand van het DB-universum VBU en de transitierelatie VBR uit Voorbeeld 5.1. We geven eerst de informele omschrijvingen, vervolgens de onderliggende queries en tenslotte de bedoelde delete-operaties.

De informele omschrijvingen van de verwijder-opdrachten luiden:

(1) Verwijder de medewerker met medewerkersnummer 3.

(2) Verwijder alle medewerkers van afdeling 3.

(3) Verwijder alle medewerkers van afdeling 3 die geen manager van een afdeling zijn.

(4) Verwijder alle afdelingen zonder medewerkers.

De queries die aan deze delete-opdrachten ten grondslag liggen zullen we gemakshalve q71, q72, q73 en q74 noemen:

(1) $q71 = \lambda v \in VBU : \{t \in v(MEDEW) \mid t(NR) = 3\}$.

(2) $q72 = \lambda v \in VBU : \{t \in v(MEDEW) \mid t(AFDNR) = 3\}$.

(3) $q73 = \lambda v \in VBU : \{t \in v(MEDEW) \mid t(AFDNR) = 3 \text{ en } t(NR) \notin \{a(MANNR) \mid a \in v(AFD)\}\}$.

(4) $q74 = \lambda v \in VBU : \{a \in v(AFD) \mid a(ANR) \notin \{t(AFDNR) \mid t \in v(MEDEW)\}\}$.

De gevraagde delete-operaties zijn nu:

(1) Del(VBU, VBR, MEDEW, q71).

(2) Del(VBU, VBR, MEDEW, q72).

(3) Del(VBU, VBR, MEDEW, q73).

(4) Del(VBU, VBR, AFD , q74).

Gezien de definitie van VBR worden delete-pogingen op de MEDEW-tabel niet beïnvloed door dynamische constraints. We gaan elk van de vier delete-operaties nog even aan een nadere beschouwing onderwerpen.

Ad(1): Hier wordt maximaal één tupel verwijderd. Er wordt uiteraard niets verwijderd als 3 niet als medewerkersnummer voorkomt. Doch ook als 3 wel als medewerkersnummer in de MEDEW-tabel voorkomt kan het zijn dat er, wegens een statische constraint, niets wordt verwijderd, namelijk in het geval dat 3 ook als managersnummer in de AFD-tabel voorkomt.

Ad(2): Hier worden nul, één of meer MEDEW-tupels "tegelijk" verwijderd. Als echter ten minste één medewerker van afdeling 3 ook een manager is (al dan niet van afdeling 3 zelf), dan wordt geen enkel MEDEW-tupel verwijderd.

Ad(3): Deze opdracht is een nuancering van de vorige. Hier worden alle in aanmerking komende MEDEW-tupels verwijderd, niet gehinderd door enige statische of dynamische constraint.

Ad(4): Uit de dynamische constraint DC1 in de definitie van de transitierelatie VBR blijkt dat er geen ANR-waarden mogen verdwijnen en, daar {ANR} een sleutel van AFD in VBU is, derhalve überhaupt geen AFD-tupels! Dit betekent dus dat elke verwijderingspoging betreffende de AFD-tabel gedoemd is te mislukken. Uit deze "intrinsieke rollback" concluderen we dat

Del(VBU, VBR, AFD, q74) = id(VBU).

□ Voorbeeld 7.4.

De laatste (en meest subtiele) klasse van gangbare transacties die we behandelen is de klasse van de zogeheten "update-operaties". Het gaat hierbij om wijzigingen van (delen van) bestaande tupels. Wat we hierbij nodig hebben is een functie q over het betreffende DB-universum U waarbij $q(v)$ voor elke $v \in U$ aangeeft *welke gedeelten* van *welke E-tupels* op *welke wijze* veranderd dienen te worden. Meer bepaald is het onze bedoeling dat $q(v)$ een functie is over een deelverzameling van $v(E)$, namelijk de verzameling van de te wijzigen E-tupels, en wel zodanig dat $q(v)\,(t)$ voor elke $t \in \mathrm{dom}(q(v))$ het gewijzigde tupel*fragment* van t weergeeft. Dit leidt ons tot de volgende algemene definitie:

DEFINITIE 7.9:
Als U een DB-universum is en $R \subseteq U \times U$ en $E \in \mathrm{He}(U)$ en q is een functie over U en $\forall v \in U : q(v)$ is een functiewaardige functie met $\mathrm{dom}(q(v)) \subseteq v(E)$, dan:

$$\mathrm{Upd}(U, R, E, q) \stackrel{D}{=} \mathrm{Main}(U, R, E, \lambda v \in U : (v(E) - \mathrm{dom}(q(v))) \cup \{t \,\theta\, q(v)\,(t) \mid t \in \mathrm{dom}(q(v))\}).$$

Met andere woorden, elk element t van $\mathrm{dom}(q(v))$ wordt vervangen door $t \,\theta\, q(v)\,(t)$. Ook hier geldt weer het voorbehoud dat na de gewenste wijzigingen aan alle constraints moet zijn voldaan; zo niet, dan blijft de toestand weer zoals het was.

VOORBEELD 7.5:

We geven enkele voorbeelden van update-operaties aan de hand het DB-universum UZKH uit Hoofdstuk 4 en de transitierelatie RZKH uit §5.2. Evenals in Voorbeeld 7.4 geven we weer eerst de informele omschrijvingen, dan de onderliggende "update-queries" en tenslotte de bedoelde update-operaties.

De informele omschrijvingen van de wijzigingsopdrachten luiden:

(1) Zet voor ieder werknemer het aantal ziektedagen weer op nul.

(2) Wijzig voor de plaats Lutjebroek het aantal tandartsen in 5, verhoog het aantal huisartsen met 2 en verdubbel het aantal apothekers.

(3) Wijzig van huisarts 99 het patiëntenaantal (per 1 januari j.l.) in 747, pas de "patiëntentoename vorig jaar" dienovereenkomstig aan (nu we een jaar verder zijn) en verhoog de verwachte toe- of afname met 10% (eventueel naar beneden afgerond).

(4) Voeg voor de patiënt met nummer 123453 een dochtertje Cherelien toe, geboren op 26 februari 1988.

(5) Verminder voor elk medicijn de voorraad met de totaal op 3 mei 1989 verstrekte hoeveelheid.

Aan deze wijzigingsopdrachten liggen de volgende queries over UZKH ten grondslag:

(1) $q75 = \lambda v \in \text{UZKH} : \lambda t \in v(\text{WN}) : \{(\text{AZD}; 0)\}$.

(2) $q76 = \lambda v \in \text{UZKH} : \lambda t \in v(\text{PLR})$ en $t(\text{WPL}) =$ 'Lutjebroek':
$\{(\text{ATA}; 5), (\text{AHA}; t(\text{AHA})+2), (\text{AAP}; 2 * t(\text{AAP}))\}$.

(3) $q77 = \lambda v \in \text{UZKH} : \lambda t \in v(\text{HA})$ en $t(\text{NRHA}) = 99$:
$\{(\text{PAT}; 747), (\text{TOEN}; 747 - t(\text{PAT})),$
$(\text{VWTN}; t(\text{VWTN}) + t(\text{VWTN}) \text{ div } 10)\}$.

(4) $q78 = \lambda v \in \text{UZKH} : \lambda t \in v(\text{PAT})$ en $t(\text{PNR}) = 123453$:
$\{(\text{KNDG}; t(\text{KNDG}) \cup \{t78\})\}$,
waarbij $t78 = \{(\text{VNM};$ 'Cherelien'$), (\text{GBD}; 19880226), (\text{GESL},$ 'V'$)\}$.

(5) $q79 = \lambda v \in \text{UZKH} : \lambda t \in v(\text{MED})$:
$\{(\text{VRD}; t(\text{VRD}) - (\Sigma x \in v(\text{MVS})$ en $x(\text{MCD}) = t(\text{MCD})$ en $x(\text{DAT}) = 890503$:
$x(\text{DUUR}) * x(\text{FREQ}) * x(\text{AEK})))\}$.

De gevraagde update-operaties zijn nu:

(1) Upd(UZKH, RZKH, WN , q75).

(2) Upd(UZKH, RZKH, PLR , q76).

(3) Upd(UZKH, RZKH, HA , q77).

(4) Upd(UZKH, RZKH, PAT , q78).

(5) Upd(UZKH, RZKH, MED, q79).

We geven bij elk van de vijf wijzigingsopdrachten nog enig commentaar. In het bijzonder gaan we "met de hand" na welke dynamische en statische constraints tot een rollback zouden kunnen leiden. (Het is echter de taak van het DBMS om deze constraints "automatisch" te controleren.)

Ad(1): Ter illustratie van Definitie 7.9 (en enkele andere in dit hoofdstuk ingevoerde begrippen) willen we voor dit eerste voorbeeld de gegeven oplossing nog even expliciet "terugrekenen". We maken daarbij gebruik van de definities 7.5 en 7.3 en de eigenschappen van q75 dat $\mathrm{dom}(q75(v)) = v(\mathrm{WN})$ en $q75(v)\,(t) = \{(\mathrm{AZD};\, 0)\}$.

$$
\begin{aligned}
&\mathrm{Upd}(\mathrm{UZKH}, \mathrm{RZKH}, \mathrm{WN}, q75)\\
&= \mathrm{Main}(\mathrm{UZKH}, \mathrm{RZKH}, \mathrm{WN}, \lambda v \in \mathrm{UZKH}: (v(\mathrm{WN}) - \mathrm{dom}\,(q75(v))) \cup\\
&\qquad\qquad\qquad\qquad\qquad\qquad \{t \,\theta\, q75(v)\,(t) \mid t \in \mathrm{dom}(q75(v))\})\\
&= \mathrm{Main}(\mathrm{UZKH}, \mathrm{RZKH}, \mathrm{WN}, \lambda v \in \mathrm{UZKH}: \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})\})\\
&= \mathrm{Ad}(\mathrm{UZKH}, \mathrm{RZKH}, \lambda v \in \mathrm{UZKH}: v \,\theta\, \{(\mathrm{WN};\, \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})\})\})\\
&= \mathrm{Id}(\mathrm{UZKH})\,\theta\,(\mathrm{RZKH} \cap \lambda v \in \mathrm{UZKH}: v \,\theta\, \{(\mathrm{WN};\, \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})
\end{aligned}
$$

Uit de definitie van UZKH in §4.2 blijkt vrij eenvoudig dat $v \,\theta\, \{(\mathrm{WN};\, \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})\})\}$ voor elke $v \in \mathrm{UZKH}$ weer een element van UZKH is. Immers, alleen de AZD-waarden van de WN-tabel worden maar aangepast, en de enige constraint die er ten aanzien van deze AZD-waarden geldt is de eis dat ze in de verzameling [0 .. 290] moeten liggen. Hieraan is in de nieuwe situatie duidelijk voldaan. Verder voldoet voor elke $v \in \mathrm{UZKH}$ het paar $(v;\, v \,\theta\, \{(\mathrm{WN};\, \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})\})\})$ aan de dynamische constraint C12 in §5.2 en (op triviale wijze) ook aan alle andere constraints in de definitie van de transitierelatie RZKH. We kunnen daarom de hierboven gegeven vereenvoudiging van Upd(UZKH, RZKH, WN, q75) nog één stap verder doorvoeren:

$$
\begin{aligned}
&\mathrm{Upd}(\mathrm{UZKH}, \mathrm{RZKH}, \mathrm{WN}, q75)\\
&= \lambda v \in \mathrm{UZKH} : v \,\theta\, \{(\mathrm{WN};\, \{t \,\theta\, \{(\mathrm{AZD};\, 0)\} \mid t \in v(\mathrm{WN})\})\}.
\end{aligned}
$$

Ad(2): Wanneer de plaats Lutjebroek niet voorkomt in v(PLR), dan zijn er geen wijzigbare tupels. De toestand blijft dan ook gewoon ongewijzigd (zonder dat er van een rollback sprake is). We bekijken nu wat er zou gebeuren wanneer de plaats Lutjebroek wel voorkomt in v(PLR). Er zijn in §5.2 geen dynamische constraints die betrekking hebben op de PLR-tabel. Bekijken we de definitie van UZKH in §4.2, dan lezen we daar uit af dat de statische (tupel)constraint R44 de enige constraint is die roet in het eten kan gooien. Wanneer namelijk de plaats Lutjebroek wel in v(PLR) voorkomt en door de verhoging van het aantal huisartsen met twee dat aantal opeens vier of meer wordt - en dus twee of drie was - én bovendien de dienstdoende huisarts Lutjebroek niet als praktijkplaats heeft, dan gaat de wijziging niet door. (Er is dan *wel* sprake van een rollback.)

Ad(3): Wanneer er geen tupel met huisartsnummer 99 voorkomt in v(HA), dan worden er geen tupels gewijzigd. We bekijken nu het effect wanneer er wel zo'n tupel in v(HA) voorkomt. De enige dynamische constraint in de definitie van de transitierelatie RZKH die betrekking heeft op de HA-tabel is C14. Het gewijzigde tupel t' in de nieuw voorgestelde toestand voldoet inderdaad aan de aldaar gestelde eis dat $t'(\text{TOEN}) = t'(\text{PAT}) - t(\text{PAT})$, want $t'(\text{PAT}) = 747$. De enige statische constraints in de definitie van het DB-universum UZKH die betrekking hebben op de te wijzigen attributen in de HA-tabel zijn de tupelconstraints R40 en R41. Voor het gewijzigde tupel t' geldt dat $t'(\text{PAT}) - t'(\text{TOEN}) = 747 - (747 - t(\text{PAT})) = t(\text{PAT})$. Aangezien het "oude" tupel t afkomstig is van een element v van UZKH zelf, voldoet t dus aan de in de objectkarakterisering FHA gestelde eis dat $t(\text{PAT}) \in \mathbb{N}$. Dus $t(\text{PAT}) \geq 0$. We concluderen hieruit dat de tupelconstraint R40 bij deze transactie dus niet kan leiden tot een rollback. Wat de tupelconstraint R41 betreft, wordt voor het gewijzigde tupel t' geëist dat $t'(\text{PAT}) + t'(\text{VWTN}) \geq 0$, ofwel dat $747 + t(\text{VWTN}) + (t(\text{VWTN}) \operatorname{div} 10) \geq 0$ voor het oude tupel t. Dit laatste komt neer op de eis dat $t(\text{VWTN}) > -680$. Wanneer dus de "oude verwachting" een daling van 680 of meer patiënten was, dan kan de voorgestelde wijziging niet gehonoreerd worden.

Ad(4): Wanneer er geen tupel met patiëntnummer 123453 voorkomt in v(PAT), dan vindt er gewoon geen wijziging plaats. Wanneer zo'n tupel wel voorkomt, dan is de enige constraint die tot een rollback kan leiden de statische (attribuut)constraint R02. Deze eis houdt in dit geval in dat onder de reeds geregistreerde kinderen van patiënt 123453 de naam Cherelien nog niet voorkomt (tenzij met geboortedatum 19880226 en geslacht 'V').

Ad(5): Hier komen alle MED-tupels voor wijziging in aanmerking. De enige constraint die bij deze transactie tot een rollback kan leiden is de statische constraint voor het attribuut VRD van de MED-tabel dat de voorraad groter dan of gelijk aan 0 moet zijn. Wanneer door de voorgestelde wijzigingen één of meer van de voorraden negatief zou worden, dan wordt van *geen enkel* medicijn de voorraad gewijzigd. (Een en ander wijst kennelijk op een administratieve fout die tot inaccurate gegevens heeft geleid, of dreigt te leiden, waardoor het blijkbaar veiliger is bevonden om dan nog maar helemaal geen veranderingen door te voeren.)

□ Voorbeeld 7.5.

OPGAVEN

7.4. In deze opgave komen we terug op Upd(UZKH, RZKH, WN, q75), de eerste van de in Voorbeeld 7.5 behandelde transacties. We zullen deze transactie even Φ noemen. Bewijs dat Φ *idempotent* is onder functiecompositie, dat wil zeggen dat $\Phi \circ \Phi = \Phi$.

7.5. Ga na welke statische en dynamische constraints tot een rollback van de transactie Inst(UZKH, RZKH, PAT, t) kunnen leiden als gegeven is dat $t \in \Pi(\text{FPAT})$. Hierbij is FPAT de in §4.2 gedefinieerde objectkarakterisering voor de tabelindex PAT.

7.6. Geef de volgende, informeel omschreven opdrachten over het DB-universum VBU voorzien van de transitierelatie VBR uit Voorbeeld 5.1 formeel weer.

(a) Geef alle medewerkers van de afdelingen 3 en 4 een salarisverhoging van 5% (waar nodig naar beneden afgerond).

(b) Geef elke manager een salarisverhoging van *f* 100,-.

(c) Geef elke manager een salarisverhoging van *f* 100,- en bovendien alle medewerkers van de afdelingen 3 en 4 een salarisverhoging van 5%. (Voor managers die aan afdeling 3 of 4 verbonden zijn, geldt deze 5% niet over deze extra *f* 100,-.)

(d) Geef elke manager een salarisverhoging van *f* 100,- voor *elke* afdeling waarvan hij of zij manager is.

7.7. Deze opgave heeft betrekking op Voorbeeld 2.13. Laat $t_A \in \Pi(\text{F5})$ en $t_B \in \Pi(\text{F5})$ met $t_A \neq t_B$ en $t_A(\text{PROV}) = t_B(\text{PROV})$.

De plaats gerepresenteerd door t_A wordt geannexeerd door de plaats gerepresenteerd door t_B. Dit betekent dat t_A de naam van t_B erft, de burgemeester van t_A gewoon burger van t_B wordt en de inwoners van t_A voortaan inwoners van t_B zijn. Geboorteplaatsen worden niet gewijzigd.

Geef een formele definitie van de transactie Annex(t_A, t_B) $\in$ U1 $\twoheadrightarrow$ U1 die de administratieve verwerking van de geschetste gemeentelijke herindeling representeert.

7.8. Geef een formele definitie van de transactie Delpb(t_0) $\in$ UZKH $\twoheadrightarrow$ UZKH, waarbij $t_0 \in \Pi(\text{FPB}) \upharpoonright \{\text{PNR, BCD, VNR}\}$, die aan elke DB-toestand $v \in$ UZKH toevoegt de toestand die uit v ontstaat door het (eventuele) patiëntbehandelingstupel met de door t_0 beschreven steutelwaarde weg te laten en bovendien de daarvoor in aanmerking komende volgnummers met 1 te verlagen. (Er zijn geen dynamische constraints.)

7.9. Geef een formele definitie van de transactie Updsous(x_0) $\in$ UZKH $\twoheadrightarrow$ UZKH binnen RZKH, waarbij $x_0 \in$ Vng(4), die aan elke DB-toestand $v \in$ UZKH toevoegt de toestand die uit v ontstaat wanneer een werknemer met identiteitsnummer x_0 de taken van zijn souschef, dat wil zeggen inclusief diens groepsleiderschap, overneemt.
(De souschef wordt een gewoon lid van zijn groep en als werknemer x_0 reeds groepsleider was dan worden die twee groepen als samengevoegd beschouwd.)

7.10. Deze opgave heeft betrekking op Voorbeeld 3.6.
Geef een formele definitie van de transactie Make(p) $\in$ UBP $\twoheadrightarrow$ UBP, waarbij $p \in$ [40000 .. 99999], die aan elke DB-toestand $v \in$ UBP toevoegt de toestand die uit v is ontstaan wanneer één exemplaar van het produkt met produktnummer p rechtstreeks is geproduceerd vanuit z'n directe onderdelen, mits p als produktnummer bekend is en alle directe onderdelen in voldoende aantallen in voorraad zijn; is dat niet het geval dan blijft de toestand dezelfde. (Er zijn geen dynamische constraints.)

7.11. Geef een formele definitie van de transactie Insmvs(x_0) $\in$ UZKH $\twoheadrightarrow$ UZKH, waarbij $x_0 \in \Pi$(FMVS $\not\div$ {VNR}), die aan elke $v \in$ UZKH toevoegt de toestand die uit v ontstaat door toevoeging van het door x_0 bijna geheel beschreven medicijnverstrekkingstupel, waarbij

- het volgnummer het eerste "vrije" positieve getal voor die patiënt is,
- ook het aantal eenheden in voorraad is aangepast, en
- er geen dynamische constraints zijn,

mits een en ander zou resulteren in een toestand die tot UZKH behoort; is dit niet het geval dan blijft de toestand zoals het was.

□

8 DATA DICTIONARIES

De *data dictionary*, ook wel *systeemcatalogus* genoemd, vormt een centraal onderdeel van een database management systeem. Het bevat *meta-gegevens*, dat wil zeggen *gegevens over de gegevens* in de database. De data dictionary kan gegevens bevatten zoals

(a) tabelnamen, attributen, de voor die attributen toegestane waarden, sleutels, verbindingseisen en andere constraints, kortom een beschrijving van het betreffende DB-universum,

(b) gegevens over de huidige DB-toestand, zoals bijvoorbeeld het aantal elementen per tabel of het aantal verschillende waarden per attribuut van een tabel,

(c) namen en definities van views, menus, procedures, schermen, schermvelden, etcetera,

(d) (statistische) gegevens over het database-gebruik tot nu toe, bijvoorbeeld uitgesplitst per afdeling en/of per periode,

(e) gegevens over de fysieke structuur van de database (unique en non-unique indexes, clusters, disk devices, gereserveerde ruimte, bezette pages per tabel),

(f) autorisatiegegevens (wie mag wat zien en wie mag wat doen?),

(g) "creators" en eigenaren van tabellen, views, programma's, etcetera,

(h) toegestane gebruikers en gebruikersgroepen, hun wachtwoorden en privileges,

(i) gegevens over lopende processen, zoals bijvoorbeeld de door het proces "gelockte" gegevens, het aantal disk reads en disk writes en de aanvragende gebruiker,

(j) gegevens over de distributie van de feitelijke gegevens over de diverse machines (in geval van een gedistribueerde database), en

(k) eventueel ook gegevens over de data dictionary zelf.

Voor meer informatie over bovengenoemde aspecten verwijzen we de lezer naar de literatuur, bijvoorbeeld naar [Da 86], [IDT 88], [NGI 88], [NGI 87], [We 86], [Ma 84], [ALM 82], [LP 82], [CD 81], [SM 80], [Uh 73] en naar de daarin genoemde referenties.

In dit hoofdstuk zullen we aan de hand van het gegevensdefinitie-gedeelte van een data dictionary, dat wil zeggen aan de hand van de onder (a) genoemde meta-gegevens, laten zien hoe een data dictionary niet alleen formeel kan worden *weergegeven*, maar ook hoe de *semantiek* ervan kan worden *geformaliseerd*. We besluiten deze inleiding met de opmerking dat dit hoofdstuk een bewerking is van [Br 88].

8.1 HET DATA-DEFINITIEGEDEELTE VAN EEN DATA DICTIONARY

Het data-definitiegedeelte van een data-dictionarysysteem bepaalt welke DB-universa met behulp van dit data-dictionarysysteem kunnen worden beschreven. Elke beschrijving die specificeerbaar is middels het betreffende data-dictionarysysteem bepaalt op eenduidige wijze een DB-universum, een wijze die specifiek is voor het systeem in kwestie. Het data-definitiegedeelte van een data-dictionarysysteem kunnen we daarom formeel representeren door een functie die aan elke (binnen dit systeem specificeerbare) beschrijving het bedoelde DB-universum toevoegt. We zullen zo'n functie een *DD-functie* noemen:

DEFINITIE 8.1:
I is een **DD-functie** $\overset{D}{\Longleftrightarrow}$ I is een functie en
$\forall d \in \mathrm{dom}(I) : I(d)$ is een DB-universum.

Als I een DD-functie is, dan noemen we dom(I) wel *het DD-universum van I*; als $U \in \mathrm{rng}(I)$, dan zeggen we wel dat *U kan worden beschreven met behulp van I*. Als bovendien $(d\,;\,U) \in I$, dan noemen we d wel een *beschrijving van U volgens I* en omgekeerd U wel *het door d volgens I beschreven DB-universum* of ook wel *de semantiek* (of *interpretatie*) *van d volgens I*.

VOORBEELD 8.1:

We beginnen met een relatief eenvoudig voorbeeld van een DD-functie. We zullen deze functie I1 noemen. Elk element van dom(I1) is een DB-skelet waarin elke tabelindex en elk attribuut een string is, en wel van ten hoogste 16 tekens. Kortom, dom(I1) = (Chs(16) $\overset{\sim}{\twoheadrightarrow}$ **P**(Chs(16))). De DD-functie I1 beeldt elke $g \in \mathrm{dom}(\mathrm{I1})$ af op het eenduidig bepaalde DB-universum over g waarin elk attribuut een willekeurige string van ten hoogste 16 tekens als waarde kan hebben en waarin verder geen andere voorwaarden gelden. Met andere woorden, voor elke tabelindex $E \in \mathrm{dom}(g)$ kan een "E-tabel"

een willekeurige deelverzameling van $g(E) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16)$ zijn. Kortom, het door g volgens I1 beschreven DB-universum is

$$\Pi\,(\lambda E \in \mathrm{dom}(g) : \mathbf{P}(g(E) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16)))$$

De bedoelde DD-functie I1 kunnen we daarom als volgt definiëren:

$$\begin{aligned} \mathrm{I1} = \lambda g \in (\mathrm{Chs}(16) \mathrel{\tilde{-\!\!\triangleright}} \mathbf{P}(\mathrm{Chs}(16))) : \\ \Pi\,(\lambda E \in \mathrm{dom}(g) : \mathbf{P}(g(E) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16))). \end{aligned}$$

Het blijkt dat er in rng(I1) per DB-skelet $g \in \mathrm{Chs}(16) \mathrel{\tilde{-\!\!\triangleright}} \mathbf{P}(\mathrm{Chs}(16))$ precies één DB-universum over g voorkomt. Dit DB-universum kan ook maar op één manier worden beschreven met behulp van I1 daar I1 een *injectieve* functie blijkt te zijn.

Ter illustratie zullen we I1 toepassen op het volgende kleine DB-skelet:

$$\begin{array}{ll|l} \mathrm{g3} = \{ & (\text{'EMP'}; \{\text{'ENAME'}, \text{'ADDRESS'}, \text{'CITY'}, \text{'DCODE'}\}), & \text{employees} \\ & (\text{'DEP'}; \{\text{'DNAME'}, \text{'DCODE'}, \text{'MANNAME'}\})\} & \text{departments} \end{array}$$

Het is eenvoudig in te zien dat g3 een element van $\mathrm{Chs}(16) \mathrel{\tilde{-\!\!\triangleright}} \mathbf{P}(\mathrm{Chs}(16))$ is. Dus g3 $\in$ dom(I1). We gaan nu uitrekenen welk DB-universum volgens I1 door g3 wordt beschreven. We berekenen het DB-universum I1(g3) door achtereenvolgens de definitie van I1, de definitie van g3 en Lemma 0.15(a) toe te passen:

$$\begin{aligned}
&\mathrm{I1(g3)} \\
&= \Pi(\lambda E \in \mathrm{dom(g3)}: \mathbf{P}(\mathrm{g3}(E) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16))) \\
&= \Pi(\{(\text{'EMP'}; \mathbf{P}(\mathrm{g3}(\text{'EMP'}) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16))), \\
&\qquad (\text{'DEP'}; \mathbf{P}(\mathrm{g3}(\text{'DEP'}) \mathrel{-\!\!\triangleright} \mathrm{Chs}(16)))\}) \\
&= \Pi(\{(\text{'EMP'}; \mathbf{P}(\Pi(\{(\text{'ENAME'} ; \mathrm{Chs}(16)), \\
&\qquad\qquad\qquad\qquad (\text{'ADDRESS'} ; \mathrm{Chs}(16)), \\
&\qquad\qquad\qquad\qquad (\text{'CITY'} ; \mathrm{Chs}(16)), \\
&\qquad\qquad\qquad\qquad (\text{'DCODE'} ; \mathrm{Chs}(16))\}))), \\
&\qquad (\text{'DEP'}; \mathbf{P}(\Pi(\{(\text{'DNAME'} ; \mathrm{Chs}(16)), \\
&\qquad\qquad\qquad\qquad (\text{'DCODE'} ; \mathrm{Chs}(16)), \\
&\qquad\qquad\qquad\qquad (\text{'MANNAME'}; \mathrm{Chs}(16))\})))\}).
\end{aligned}$$

Waar we ons in de voorgaande hoofdstukken niet of nauwelijks hoefden uit te laten over de aard van tabelindexen en attributen, zullen we in de voorbeelden in dit hoofdstuk voor tabelindexen en attributen telkens expliciet kiezen voor strings. Dientengevolge gaan we daarom nu ook zorgvuldiger met quotes om.

□ Voorbeeld 8.1.

We noemen een DD-functie I *ten minste even expressief als* een DD-functie I' als met I alle DB-universa kunnen worden beschreven die ook met I' kunnen worden beschreven.
We noemen I en I' *DD-equivalent* als ze precies dezelfde verzameling DB-universa kunnen beschrijven. Formeel:

DEFINITIE 8.2:
Als I en I' DD-functies zijn, dan:

(a) **I is ten minste even expressief als I'** $\overset{D}{\Longleftrightarrow} \mathrm{rng}(I') \subseteq \mathrm{rng}(I)$;

(b) **I is DD-equivalent met I'** $\overset{D}{\Longleftrightarrow} \mathrm{rng}(I') = \mathrm{rng}(I)$.

Twee DD-equivalente DD-functies kunnen niet alleen verschillen in hun domein, dus in hun verzameling beschrijvingen, maar ook in de manier waarop hun eventuele *gemeenschappelijke* beschrijvingen worden geïnterpreteerd: als I en I' twee DD-equivalente DD-functies zijn en $d \in \mathrm{dom}(I) \cap \mathrm{dom}(I')$, dan hoeft $I(d)$ niet gelijk te zijn aan $I'(d)$. Omgekeerd hoeven twee DD-functies met hetzelfde domein niet DD-equivalent te zijn, zie de voorbeelden 8.4 en 8.5.

We merken op dat de range van een DD-functie I een *verzameling* DB-universa is en daarom niet de *klasse* van *alle* DB-universa omvat. (Voor een uitgebreide behandeling van dit verzamelingstheoretische onderscheid verwijzen we de lezer naar bijvoorbeeld [DDS 75].) Er bestaat daarom voor elke DD-functie I wel een DD-functie I' die expressiever is, dat wil zeggen waarmee "meer" DB-universa kunnen worden beschreven dan met I. We komen dan ook tot de (commercieel) belangwekkende conclusie dat er dus niet zoiets bestaat als een "universele, alomvattende" DD-functie waarmee de klasse van *alle* DB-universa zou kunnen worden beschreven.

Het volgende lemma suggereert een eenvoudige manier om de expressiviteit van twee DD-functies met elkaar te vergelijken.

LEMMA 8.1:
Als I en I' DD-functies zijn en h is een functie zodanig dat $I' = I \circ h$,
dan is I ten minste even expressief als I'.

Dit lemma volgt direct uit Definitie 8.2(a) en Lemma 0.9(c). De functie h "converteert" een beschrijving volgens I' naar een beschrijving volgens I. We noemen h in dit verband dan ook wel een *DD-conversiefunctie* of, als h relatief "eenvoudig" is, een *DD-migratiefunctie*.

VOORBEELD 8.2:

We zullen een DD-functie I2 definiëren die DD-equivalent is met de DD-functie I1 uit Voorbeeld 8.1. In essentie is I2 een "opgepoetste" versie van I1 waarin het DD-universum in de vorm van een *DB-universum* is gegoten. Het DD-universum van I2, dat wil zeggen de verzameling mogelijke beschrijvingen volgens I2, is een DB-universum DDU2 met slechts één "meta-tabelindex", de string 'ATT', en twee "meta-attributen", de strings 'TNAAM' en 'ANAAM'. De verzameling van de "object-tabelindexen" van het DB-universum beschreven door een element d van dom(I2) is $\{t(\text{'TNAAM'}) \mid t \in d(\text{'ATT'})\}$ en de verzameling attributen van zo'n "object-tabelindex" E is $\{x(\text{'ANAAM'}) \mid x \in d(\text{'ATT'}) \text{ en } x(\text{'TNAAM'}) = E\}$.
De definities van DDU2 en I2 luiden als volgt:

$$\text{DDU2} = \Pi(\{(\text{'ATT'} ; \mathbf{P}(\{\text{'TNAAM'}, \text{'ANAAM'}\} \twoheadrightarrow \text{Chs}(16)))\});$$

$$\begin{aligned} \text{I2} = \lambda d \in \text{DDU2}&: \\ &\Pi(\lambda E \in \{t(\text{'TNAAM'}) \mid t \in d(\text{'ATT'})\}: \\ &\qquad \mathbf{P}(\{x(\text{'ANAAM'}) \mid x \in d(\text{'ATT'}) \text{ en } x(\text{'TNAAM'}) = E\} \twoheadrightarrow \text{Chs}(16))) \end{aligned}$$

Het door g3 volgens I1 beschreven DB-universum uit Voorbeeld 8.1 kan ook worden beschreven volgens I2, en wel door de functie d1 over {'ATT'} waarvoor d1('ATT') de in Figuur 8.1 gerepresenteerde tabel over {'TNAAM', 'ANAAM'} is.

d1('ATT'):

TNAAM	ANAAM
EMP	ENAME
EMP	ADDRESS
EMP	CITY
EMP	DCODE
DEP	DCODE
DEP	DNAME
DEP	MANNAME

Figuur 8.1: Een beschrijving volgens I2

Ter illustratie van de definitie van I2 zullen we I2(d1) even uitrekenen. Bij de bereke-

ning van I2(d1) maken we achtereenvolgens gebruik van de definitie van I2, tweemaal van de definitie van d1 en tenslotte van Lemma 0.15(a):

$$
\begin{aligned}
&\mathrm{I2(d1)}\\
&= \Pi(\lambda E \in \{t(\text{'NAAM'}) \mid t \in \mathrm{d1}(\text{'ATT'})\}:\\
&\qquad\qquad \mathrm{P}(\{x(\text{'ANAAM'}) \mid x \in \mathrm{d1}(\text{'ATT'}) \text{ en } x(\text{'TNAAM'}) = E\} \twoheadrightarrow \mathrm{Chs}(16)))\\
&= \Pi(\lambda E \in \{\text{'EMP'}, \text{'DEP'}\}:\\
&\qquad\qquad \mathrm{P}(\{x(\text{'ANAAM'}) \mid x \in \mathrm{d1}(\text{'ATT'}) \text{ en } x(\text{'TNAAM'}) = E\} \twoheadrightarrow \mathrm{Chs}(16)))\\
&= \Pi(\{(\text{'EMP'}; \mathrm{P}(\{\text{'ENAME'}, \text{'ADDRESS'}, \text{'CITY'}, \text{'DCODE'}\} \twoheadrightarrow \mathrm{Chs}(16))),\\
&\qquad (\text{'DEP'}; \mathrm{P}(\{\text{'DNAME'}, \text{'DCODE'}, \text{'MANNAME'}\} \twoheadrightarrow \mathrm{Chs}(16)))\})\\
&= \Pi(\{(\text{'EMP'}; \mathrm{P}(\Pi(\{(\text{'ENAME'} \;;\ \mathrm{Chs}(16)),\\
&\qquad\qquad\qquad\qquad (\text{'ADDRESS'} \;;\ \mathrm{Chs}(16)),\\
&\qquad\qquad\qquad\qquad (\text{'CITY'} \;;\ \mathrm{Chs}(16)),\\
&\qquad\qquad\qquad\qquad (\text{'DCODE'} \;;\ \mathrm{Chs}(16))\}))),\\
&\qquad (\text{'DEP'}; \mathrm{P}(\Pi(\{(\text{'DNAME'} \;;\ \mathrm{Chs}(16)),\\
&\qquad\qquad\qquad\qquad (\text{'DCODE'} \;;\ \mathrm{Chs}(16)),\\
&\qquad\qquad\qquad\qquad (\text{'MANNAME'}; \mathrm{Chs}(16))\})))\}).
\end{aligned}
$$

We zien dus dat I2(d1) = I1(g3).

Dat het volgens I1 beschreven DB-universum aan het eind van Voorbeeld 8.1 ook kan worden beschreven volgens I2 is geen toevalligheid, want I2 is DD-equivalent met I1. Volgens Lemma 8.1 kan dit worden bewezen door twee conversiefuncties h12 en h21 te definiëren zodanig dat

$$\mathrm{I1} = \mathrm{I2} \circ \mathrm{h12} \;\text{ en }\; \mathrm{I2} = \mathrm{I1} \circ \mathrm{h21}$$

We definiëren hier alleen de conversiefunctie h21. De definitie van h12 en de bewijzen van de twee bovengenoemde eigenschappen worden in Opgave 8.3 aan de lezer overgelaten.

$$
\begin{aligned}
\mathrm{h21} = \lambda d \in \mathrm{DDU2}:&\\
&\lambda E \in \{t(\text{'TNAAM'}) \mid t \in d(\text{'ATT'})\}:\\
&\qquad \{x(\text{'ANAAM'}) \mid x \in d(\text{'ATT'}) \text{ en } x(\text{'TNAAM'}) = E\}
\end{aligned}
$$

□ Voorbeeld 8.2.

We merken op dat de DD-functie I2 uit Voorbeeld 8.2 de interessante eigenschap heeft dat het DD-universum van I2 zelf een DB-universum is. De eigenschap dat het *DD*-universum dezelfde "structuur" heeft als de *DB*-universa, kan in de praktijk zeer nuttig zijn. Het raadplegen en wijzigen van de data dictionary zou dan namelijk op dezelfde manier kunnen gebeuren als het raadplegen en wijzigen van de databases zelf! En aangezien het *wijzigen* van de data dictionary (alias "systeem-database") in feite neerkomt op het *definiëren* (of het herdefiniëren) van de eigenlijke database (alias "object-database"), kunnen we dus ook inzien dat definitie én gebruik van de database met hetzelfde mechanisme zouden kunnen worden gerealiseerd. (Onder *gebruik* verstaan we raadpleging en wijziging van de inhoud.)

DD-functies zoals I2, waarvan het DD-universum zelf een DB-universum is, zullen we *DB-vormig* noemen:

DEFINITIE 8.3:
Als I een DD-functie is, dan:
I is **DB-vormig** $\overset{D}{\Longleftrightarrow}$ dom(I) is een DB-universum.

Als I een DB-vormige DD-functie is dan noemen we het DD-universum dom(I) wel een *meta DB-universum*, een element van dom(I) wel een *meta DB-toestand*, de tabelindexen van het DB-universum dom(I) wel *meta-tabelindexen* en de attributen van dom(I) wel *meta-attributen* (zoals we overigens in Voorbeeld 8.2 reeds deden).

Van een DB-vormige DD-functie I kunnen we ons afvragen of het DB-universum dom(I) zelf kan worden beschreven met behulp van I, met andere woorden of er een $d_0 \in$ dom(I) bestaat zodanig dat $I(d_0) =$ dom(I), dus kortom, of dom(I) $\in$ rng(I). In dat geval noemen we *I zelfregistreerbaar*:

DEFINITIE 8.4:
Als I een DD-functie is, dan:
I is **zelfregistreerbaar** $\overset{D}{\Longleftrightarrow}$ dom(I) $\in$ rng(I).

Dat elke zelfregistreerbare DD-functie DB-vormig is, volgt onmiddellijk uit bovenstaande definitie (en is daarom niet expliciet als voorwaarde in de definitie zelf opgenomen).

Zelfregistreerbaarheid van een DD-functie I impliceert in feite dat de in *I gebruikte* structuren niet complexer zijn dan de met behulp van *I te beschrijven* structuren. Een van de interessante aspecten van zelfregistreerbare DD-functies is dat ze als het ware replica's van hun eigen DD-universum kunnen maken. Deze eigenschap kan zowel bij het bouwen van DBMS-en als voor het simuleren van de data dictionary van een gegeven DBMS zeer nuttig zijn.

VOORBEELD 8.3:

De DD-functie I2 uit Voorbeeld 8.2 is zelfregistreerbaar. Om dit aan te tonen definiëren we de volgende tabelwaardige functie:

$$d2 = \{(\text{'ATT'}; \{ \{(\text{'TNAAM'}; \text{'ATT'}), (\text{'ANAAM'}; \text{'TNAAM'})\}, \{(\text{'TNAAM'}; \text{'ATT'}), (\text{'ANAAM'}; \text{'ANAAM'})\}\})\}$$

Deze meta DB-toestand d2 kunnen we dus ook als volgt representeren:

d2('ATT'):

TNAAM	ANAAM
ATT	TNAAM
ATT	ANAAM

Figuur 8.2: Een beschrijving volgens I2

Met behulp van de definities van I2 en d2 gaan we nu I2(d2) uitrekenen:

$$\begin{aligned} I2(d2) &= \Pi(\lambda E \in \{\text{'ATT'}\} : \mathbf{P}(\{\text{'TNAAM'}, \text{'ANAAM'}\} \twoheadrightarrow Chs(16))) \\ &= \Pi(\quad \{(\text{'ATT'}; \mathbf{P}(\{\text{'TNAAM'}, \text{'ANAAM'}\} \twoheadrightarrow Chs(16)))\}) \\ &= DDU2 \\ &= dom(I2) \end{aligned}$$

Hiermee hebben we bewezen dat I2 inderdaad zelfregistreerbaar is.

De DD-functie I1 uit Voorbeeld 8.1 is niet zelfregistreerbaar, want I1 is zelfs niet DB-vormig:

$$dom(I1) = Chs(16) \twoheadrightarrow \mathbf{P}(Chs(16)),$$

dus niet alle elementen van dom(I1) hebben hetzelfde domein (wat bij een DB-universum wel het geval is).

□ Voorbeeld 8.3.

We zullen nu een wat groter (en realistischer) voorbeeld van een DD-functie geven.

VOORBEELD 8.4:

We zullen een (DB-vormige) DD-functie I3 definiëren waarmee DB-universa met de volgende mogelijkheden kunnen worden beschreven:

(M1) Attributen kunnen "van type *integer*" of "van type *string*" zijn (aangegeven met 'N' respectievelijk 'C').

(M2) De maximale "veldlengte" kan *per attribuut* worden gekozen (met een bovengrens van 256). We merken op dat in de voorbeelden 8.1 en 8.2 de maximale veldlengte in de definitie van een DB-universum niet kan worden gekozen, maar altijd 16 moet zijn. (Desalniettemin kunnen de actuele veldlengten in de tupels van een *DB-toestand* wel kleiner dan 16 zijn.)

(M3) De veldlengte van een attribuut "van type integer" kan echter niet meer dan 9 bedragen.

(M4) Een tabel(-index) kan meerdere sleutels hebben, maar elke sleutel is enkelvoudig (dat wil zeggen bestaat uit precies één attribuut).

(M5) Er kunnen verbindingseisen (van precies één attribuut naar precies één attribuut) worden weergegeven; de attributen mogen hierbij in dezelfde tabel voorkomen.

De definitie van DDU3, het DD-universum van I3, wordt voorafgegaan door de definities van twee "meta-objectkarakteriseringen", FATT3 en FREF3, een "meta-tupelkarakterisering" TATT3 en een "DD-karakterisering" DDK3 die aan elk van de twee meta-tabelindexen 'ATT' en 'REF' de verzameling toegestane "meta-tabellen" voor die meta-tabelindex toevoegt:

$$
\begin{aligned}
\text{FATT3} = \{ & (\text{'TNAME'} \; ; \text{Chs}(18)), \\
& (\text{'ANAME'} \; ; \text{Chs}(16)), \\
& (\text{'TYPE'} \; ; \{\text{'C'}, \text{'N'}\}), && \text{(M1)} \\
& (\text{'SIZE'} \; ; [1 .. 256]), && \text{(M2)} \\
& (\text{'KEY'} \; ; \{\text{'*'}, \text{'–'}\})\};
\end{aligned}
$$

$$
\begin{aligned}
\text{TATT3} = \{ t \mid\ & t \in \Pi(\text{FATT3}) \text{ en} \\
& \text{als } t(\text{'TYPE'}) = \text{'N'} \text{ dan } t(\text{'SIZE'}) \leq 9\}; && \text{(M3)}
\end{aligned}
$$

$$\begin{aligned}\text{FREF3} = \{&(\text{'TNAME1'} ; \text{Chs}(18)),\\ &(\text{'ANAME1'} ; \text{Chs}(16)),\\ &(\text{'TNAME2'} ; \text{Chs}(18)),\\ &(\text{'ANAME2'} ; \text{Chs}(16))\};\end{aligned}$$

$$\begin{aligned}\text{DDK3} = \{&(\text{'ATT'} ; \{T \subseteq \text{TATT3} \mid \{\text{'TNAME'}, \text{'ANAME'}\} \text{ is u.i. in } T\}),\\ &(\text{'REF'} ; \mathbf{P}(\Pi(\text{FREF3})))\};\end{aligned}$$

$$\begin{aligned}\text{DDU3} = \{d \mid d \in \Pi(\text{DDK3}) \text{ en }\\ &\{(\text{'TNAME1'} ; \text{'TNAME'}), (\text{'ANAME1'} ; \text{'ANAME'})\} \text{ verbindt}\\ &\quad d(\text{'REF'}) \text{ met } d(\text{'ATT'}) \text{ en}\\ &\{(\text{'TNAME2'} ; \text{'TNAME'}), (\text{'ANAME2'} ; \text{'ANAME'})\} \text{ verbindt}\\ &\quad d(\text{'REF'}) \text{ met } d(\text{'ATT'})\}.\end{aligned}$$

We definiëren nu onze DD-functie I3 over DDU3 op de volgende (top-down) manier:

$$\begin{aligned}\text{I3} = \lambda d \in \text{DDU3}:\\ &\{v \mid v \in \Pi(\text{DBK3}_d) \text{ en}\\ &\quad \forall y \in d(\text{'REF'}) : \{(y(\text{'ANAME1'}) ; y(\text{'ANAME2'}))\} \text{ verbindt}\\ &\qquad v(y(\text{'TNAME1'})) \text{ met } v(y(\text{'TNAME2'}))\},\end{aligned} \tag{M5}$$

waarbij we voor elke $d \in$ DDU3 de databasekarakterisering DBK3_d als volgt definiëren:

$$\begin{aligned}&\text{DBK3}_d =\\ &\lambda E \in \{t(\text{'TNAME'}) \mid t \in d(\text{'ATT'})\}:\\ &\quad \{T \mid T \subseteq \Pi(\text{FK}_{d,E}) \text{ en}\\ &\qquad \forall x \in d(\text{'ATT'}): \text{als } x(\text{'KEY'}) = \text{'*'} \text{ en } x(\text{'TNAME'}) = E\\ &\qquad\qquad \text{dan } \{x(\text{'ANAME'})\} \text{ is u.i. in } T\},\end{aligned} \tag{M4}$$

waarbij we voor elke $d \in$ DDU3 en elke $E \in \{t(\text{'TNAME'}) \mid t \in d(\text{'ATT'})\}$ de objectkarakterisering $\text{FK}_{d,E}$ voor E als volgt definiëren:

$$\begin{aligned}&\text{FK}_{d,E} =\\ &\{(t(\text{'ANAME'}); \text{Chs}(t(\text{'SIZE'}))) \mid t \in d(\text{'ATT'}) \text{ en } t(\text{'TNAME'}) = E \text{ en } t(\text{'TYPE'}) = \text{'C'}\} \cup\\ &\{(t(\text{'ANAME'}); \text{Int}(t(\text{'SIZE'}))) \mid t \in d(\text{'ATT'}) \text{ en } t(\text{'TNAME'}) = E \text{ en } t(\text{'TYPE'}) = \text{'N'}\}\end{aligned}$$

Figuur 8.3 representeert een beschrijving d3 volgens I3 van een iets verfijnder DB-universum dan I1(g3) uit Voorbeeld 8.1 (ofwel I2(d1) uit Voorbeeld 8.2).

d3('ATT'):

TNAME	ANAME	KEY	TYPE	SIZE
EMP	ENAME	*	C	16
EMP	ADDRESS	–	C	16
EMP	CITY	–	C	16
EMP	DCODE	–	C	16
DEP	DCODE	*	C	16
DEP	DNAME	*	C	16
DEP	MANNAME	–	C	16

d3('REF'):

TNAME1	ANAME1	TNAME2	ANAME2
EMP	DCODE	DEP	DCODE
DEP	MANNAME	EMP	ENAME

Figuur 8.3: Een beschrijving volgens I3

Ter illustratie van de definities van I3, $DBK3_d$ en $FK_{d,E}$ gaan we I3(d3) eens helemaal uitrekenen. We werken eerst I3 en dan d3('REF') uit:

I3(d3)
$= \{v \mid v \in \Pi(DBK3_{d3})$ en
$\quad \forall y \in$ d3('REF'): $\{(y(\text{'ANAME1'}); y(\text{'ANAME2'}))\}$ verbindt
$\quad\quad v(y(\text{'TNAME1'}))$ met $v(y(\text{'TNAME2'}))\}$
$= \{v \mid v \in \Pi(DBK3_{d3})$ en
$\quad \{(\text{'DCODE'}; \text{'DCODE'})\}$ verbindt $v(\text{'EMP'})$ met $v(\text{'DEP'})$ en
$\quad \{(\text{'MANNAME'}; \text{'ENAME'})\}$ verbindt $v(\text{'DEP'})$ met $v(\text{'EMP'})\}$,

waarbij (volgens de definitie van $DBK3_{d3}$ en de eigenschappen van d3('ATT'))

$$
\begin{aligned}
&\mathrm{DBK3}_{\mathrm{d3}}\\
&= \lambda E \in \{t(\text{'NAME'}) \mid t \in \mathrm{d3}\,(\text{'ATT'})\}:\\
&\qquad \{T \mid T \subseteq \Pi(\mathrm{FK}_{\mathrm{d3},E}) \text{ en}\\
&\qquad\qquad \forall x \in \mathrm{d3}\,(\text{'ATT'}):\ \text{als } x(\text{'KEY'}) = \text{'*'} \text{ en } x(\text{'TNAME'}) = E\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{dan } \{x(\text{'ANAME'})\} \text{ is u.i. in } T\}\\
&= \lambda E \in \{\text{'EMP'}, \text{'DEP'}\}:\\
&\qquad \{T \mid T \subseteq \Pi(\mathrm{FK}_{\mathrm{d3},E}) \text{ en}\\
&\qquad\qquad \forall x \in \mathrm{d3}\,(\text{'ATT'}):\ \text{als } x(\text{'KEY'}) = \text{'*'} \text{ en } x(\text{'TNAME'}) = E\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{dan } \{x(\text{'ANAME'})\} \text{ is u.i. in } T\}\\
&= \{(\text{'EMP'};\ \{T \mid T \subseteq \Pi(\mathrm{FK}_{\mathrm{d3},\,\text{'EMP'}}) \text{ en}\\
&\qquad\qquad\qquad \{\text{'ENAME'}\} \text{ is u.i. in } T\}),\\
&\qquad (\text{'DEP'};\ \{T \mid T \subseteq \Pi(\mathrm{FK}_{\mathrm{d3},\,\text{'DEP'}}) \text{ en}\\
&\qquad\qquad\qquad \{\text{'DCODE'}\} \text{ is u.i. in } T \text{ en}\\
&\qquad\qquad\qquad \{\text{'DNAME'}\} \text{ is u.i. in } T\})\},
\end{aligned}
$$

waarbij (volgens de definitie van $\mathrm{FK}_{\mathrm{d3},E}$ en het gegeven dat er in d3('ATT') alleen maar het type 'C' voorkomt)

$$
\begin{aligned}
&\mathrm{FK}_{\mathrm{d3},E}\\
&= \{(t(\text{'ANAME'});\ \mathrm{Chs}(t(\text{'SIZE'}))) \mid t \in \mathrm{d3}(\text{'ATT'}) \text{ en } t(\text{'TNAME'}) = E \text{ en}\\
&\qquad\qquad\qquad\qquad\qquad\qquad t(\text{'TYPE'}) = \text{'C'}\} \cup\\
&\quad \{(t(\text{'ANAME'});\ \mathrm{Int}(t(\text{'SIZE'}))) \mid t \in \mathrm{d3}(\text{'ATT'}) \text{ en } t(\text{'TNAME'}) = E \text{ en}\\
&\qquad\qquad\qquad\qquad\qquad\qquad t(\text{'TYPE'}) = \text{'N'}\}\\
&= \{(t(\text{'ANAME'});\ \mathrm{Chs}(16)) \mid t \in \mathrm{d3}(\text{'ATT'}) \text{ en } t(\text{'TNAME'}) = E\}
\end{aligned}
$$

zodat volgens de definitie van d3('ATT')

$$
\begin{aligned}
\mathrm{FK}_{\mathrm{d3},\,\text{'EMP'}} = \{&(\text{'ENAME'}\ ;\ \mathrm{Chs}(16)),\\
&(\text{'ADDRESS'}\ ;\ \mathrm{Chs}(16)),\\
&(\text{'CITY'}\ ;\ \mathrm{Chs}(16)),\\
&(\text{'DCODE'}\ ;\ \mathrm{Chs}(16))\} \quad \text{en}\\
\mathrm{FK}_{\mathrm{d3},\,\text{'DEP'}} = \{&(\text{'DCODE'}\ ;\ \mathrm{Chs}(16)),\\
&(\text{'DNAME'}\ ;\ \mathrm{Chs}(16)),\\
&(\text{'MANNAME'};\ \mathrm{Chs}(16))\}.
\end{aligned}
$$

Hiermee hebben we het DB-universum I3(d3) helemaal uitgerekend. Na vergelijking met Voorbeeld 8.2 is het eenvoudig in te zien dat I3(d3) $\subset$ I2(d1); immers,

$$I2(d1) = \Pi(\{(\text{'EMP'} ; \mathbf{P}(\Pi(FK_{d3,} \text{'}_{EMP}\text{'}))),\\ (\text{'DEP'} ; \mathbf{P}(\Pi(FK_{d3,} \text{'}_{DEP}\text{'})))\}).$$

Het DB-universum I3(d3) is dus duidelijk een verfijning van het constraints-arme DB-universum I2(d1).

I3 is een voorbeeld van een DB-vormige DD-functie want het domein van I3, DDU3, is een DB-universum (en wel over {('ATT'; dom(FATT3)), ('REF'; dom(FREF3))}). I3 is echter niet zelfregistreerbaar: met behulp van I3 kunnen we géén "enumeratie-typen" (zoals {'*', '–'}) beschrijven, géén willekeurige "subrange-typen" (zoals [1 .. 256]), géén tupelconstraints (zoals (M3) in de definitie van TATT3), géén samengestelde sleutels (zoals de sleutel {'TNAME', 'ANAME'} van 'ATT' in DDU3), en géén verbindingseisen met meer dan één attributenpaar (zoals die in de definitie van DDU3 zelf tot tweemaal toe voorkomen).

Het DB-universum SPSU uit Voorbeeld 2.17 kan niet worden beschreven volgens I3 omdat middels I3 geen samengestelde sleutels kunnen worden weergegeven.

De DD-functie I3 is ten minste even expressief als de DD-functie I2 uit Voorbeeld 8.2. Om dit te bewijzen is het volgens Lemma 8.1 voldoende om een functie h te definiëren zodanig dat $I2 = I3 \circ h$. De volgende conversiefunctie h23 heeft die gewenste eigenschap (zie Opgave 8.5):

$$h23 = \lambda d \in DDU2 :\\ \{(\text{'ATT'} ; \{t \cup t23 \mid t \in d(\text{'ATT'})\}),\\ (\text{'REF'} ; \varnothing)\} ,$$

waarbij $t23 = \{(\text{'TYPE'} ; \text{'C'}), (\text{'SIZE'} ; 16), (\text{'KEY'} ; \text{'–'})\}$

Dus h23 zet elke beschrijving d volgens I2 om in de (equivalente) beschrijving volgens I3 waarvan de 'REF'-tabel leeg is en de attributen 'TYPE', 'SIZE' en 'KEY' in *elk* tupel van de (nieuwe) 'ATT'-tabel de (vaste) waarden 'C', 16 en '–' hebben.

□ Voorbeeld 8.4.

Het volgende voorbeeld toont aan dat er voor een element van een gegeven DD-universum soms meer dan één "natuurlijke" interpretatie mogelijk is.

VOORBEELD 8.5:

We definiëren een tweede DD-functie over DDU3, het DD-universum waarover we in Voorbeeld 8.4 ook al de DD-functie I3 hebben gedefinieerd. Met deze nieuwe DD-functie kunnen DB-universa worden beschreven met nagenoeg dezelfde mogelijkheden als die in Voorbeeld 8.4; alleen mogelijkheid (M4) is vervangen, en wel door de volgende:

(M4′) Elke tabelindex kan maar één sleutel hebben; deze sleutel mag echter samengesteld zijn.

Dankzij het feit dat we onze definities vrij "modulair" hebben opgezet, hoeven we maar één nieuwe hulpdefinitie in te voeren om onze nieuwe DD-functie I4 te kunnen definiëren. In de hulpfunctie $\mathrm{DBK4}_d$, die aan elke tabelindex de verzameling van alle voor die tabelindex toegestane tabellen toevoegt, wordt tot uitdrukking gebracht dat voor elke tabelindex E de verzameling van alle attributen van E die van een ster ('*') zijn voorzien tezamen één sleutel vormen, althans, als er ten minste één "ster-attribuut" van E bestaat.

De definitie van onze DD-functie I4 over DDU3 luidt nu als volgt:

$$
\begin{array}{l}
\mathrm{I4} = \\
\lambda d \in \mathrm{DDU3}: \\
\qquad \{v \mid v \in \Pi(\mathrm{DBK4}_d) \text{ en} \\
\qquad\qquad \forall y \in d(\text{'REF'}) : \{(y(\text{'ANAME1'}) ; y(\text{'ANAME2'}))\} \text{ verbindt} \qquad \text{(M5)} \\
\qquad\qquad\qquad v(y(\text{'TNAME1'}) \text{ met } v(y(\text{'TNAME2'}))\}\,,
\end{array}
$$

waarbij we voor elke $d \in \mathrm{DDU3}$ de databasekarakterisering $\mathrm{DBK4}_d$ als volgt definiëren:

$$
\begin{array}{l}
\mathrm{DBK4}_d = \\
\lambda E \in \{t(\text{'TNAME'}) \mid t \in d(\text{'ATT'})\} : \\
\qquad \{T \mid T \subseteq \Pi(\mathrm{FK}_{d,E}) \text{ en} \\
\qquad\qquad \text{als } (\exists x \in d(\text{'ATT'}) : x(\text{'KEY'}) = \text{'*'} \text{ en } x(\text{'TNAME'}) = E) \qquad \text{(M4')} \\
\qquad\qquad \text{dan } \{x(\text{'ANAME'}) \mid x \in d(\text{'ATT'}) \text{ en } x(\text{'KEY'}) = \text{'*'} \text{ en} \\
\qquad\qquad\qquad x(\text{'TNAME'}) = E\} \text{ is u.i. in } T\}.
\end{array}
$$

Ter illustratie representeert Figuur 8.4 een beschrijving d4 (volgens I4) van het DB-universum SPSU uit Voorbeeld 2.17. We wijzen hierbij nog eens op het verschil in

interpretatie van de sterretjes ('*') in Voorbeeld 8.4 en die in het huidige voorbeeld.

d4('ATT'):

TNAME	ANAME	KEY	TYPE	SIZE
S	S#	*	C	5
S	SNAME	–	C	20
S	STATUS	–	N	4
S	CITY	–	C	15
P	P#	*	C	6
P	PNAME	–	C	20
P	COLOR	–	C	6
P	WEIGHT	–	N	4
P	CITY	–	C	15
SP	S#	*	C	5
SP	P#	*	C	6
SP	QTY	–	N	9

d4('REF'):

TNAME1	ANAME1	TNAME2	ANAME2
SP	S#	S	S#
SP	P#	P	P#

Figuur 8.4: Een beschrijving van SPSU volgens I4

Omdat I4 niet meer dan één sleutel per tabelindex kan beschrijven, kan het DB-universum I3(d3) uit Voorbeeld 8.4 niet worden beschreven met I4.

□ Voorbeeld 8.5.

8.2 SAMENVATTING VAN DE VOORBEELDEN

Ter afsluiting van dit hoofdstuk hebben we in Figuur 8.5 nog eens samengevat welke DD-functies welke DB-universa kunnen beschrijven en welke DD-functies DB-vormig dan wel zelfregistreerbaar zijn.

DB-un. \ DD-fu.	I1	I2	I3	I4
I1(g3) (Vb. 8.1)	+	+	+	+
I3(d3) (Vb. 8.4)	-	-	+	-
SPSU (Vb. 8.5)	-	-	-	+
DDU2 (Vb. 8.2)	+	+	+	+
DDU3 (Vb. 8.4)	-	-	-	-
DB-vormig	-	+	+	+
zelfregistreerbaar	-	+	-	-

Figuur 8.5: Samenvatting van de voorbeelden

OPGAVEN

8.1. Geef, indien mogelijk, van elk van de volgende DB-universa een beschrijving volgens I1, I2, I3 of I4; geef anders, indien mogelijk, met behulp van I3 of I4 een "benadering" van het gewenste DB-universum (in de vorm van een bij voorkeur zo klein mogelijk "omhullend" DB-universum). Als ook dat niet mogelijk is, geef daarvoor dan de reden(en) aan en probeer alsnog een "zo goed mogelijke" benadering te geven.

(a) VBU uit Voorbeeld 1.3.

(b) Uw VBU2 uit Opgave 1.11.

(c) Uw VBU3 uit Opgave 1.12.

8.2. Bewijs dat elk DB-universum dat met behulp van de DD-functie I1 uit Voorbeeld 8.1 te beschrijven is ook "de lege toestand" bevat.
(Dus te bewijzen dat $\forall g \in \mathrm{dom}(\mathrm{I1}) : (\lambda E \in \mathrm{dom}(g) : \varnothing) \in \mathrm{I1}(g)$.)

8.3. Deze opgave heeft betrekking op Voorbeeld 8.2.

(a) Bereken h21(d1).

(b) Bewijs dat $I2 = I1 \circ h21$.

(c) Definieer een functie h12 zodanig dat $I1 = I2 \circ h12$ (en bewijs deze eigenschap).

8.4. Ga voor elke $F \in \{I1, I2, I3, I4\}$ het volgende na.

(a) Is elke $U \in \mathrm{rng}(F)$ consistent?

(b) Is elke $U \in \mathrm{rng}(F)$ regulier? (Hint: $\forall n \in \mathbb{N} : (\mathrm{Chs}(n) \cap \mathbb{Z}) = \varnothing$.)

Geef telkens een bewijs of een tegenvoorbeeld.
(Zie Opgave 1.5 voor de gebruikte terminologie.)

8.5. Bewijs dat in Voorbeeld 8.4 het volgende geldt:

(a) $\forall d \in \mathrm{DDU2} : \mathrm{h23}(d) \in \mathrm{DDU3}$;

(b) $I2 = I3 \circ h23$.

8.6. Toon aan dat de DD-functie I4 uit Voorbeeld 8.5 niet injectief is, met andere woorden dat er DB-universa zijn die op meer dan één wijze met behulp van I4 kunnen worden beschreven. (Hint: let op de sleutelbeschrijvingen.)

8.7. (a) Is de DD-functie I4 uit Voorbeeld 8.5 DB-vormig?
(b) Is I4 zelfregistreerbaar?
Beargumenteer uw antwoorden.

8.8. In deze opgave vergelijken we de beschrijvingskracht van de DD-functies I3 en I4 uit de voorbeelden 8.4 en 8.5.

(a) Zijn er DB-universa die zowel volgens I3 als volgens I4 kunnen worden beschreven?

(b) Is I3 ten minste even expressief als I4?

(c) Is I4 ten minste even expressief als I3?

Geef telkens een bewijs of een geschikt tegenvoorbeeld.

8.9. Bewijs dat I3 ∘ h23 = I4 ∘ h23, waarbij h23 de in Voorbeeld 8.4 gedefinieerde conversiefunctie is.

8.10. Bewijs dat I4 uit Voorbeeld 8.5 ten minste even expressief is als I2 uit Voorbeeld 8.2.

8.11. (a) Geef een beschrijving van DDU2 (uit Voorbeeld 8.2) volgens I4.

(b) Is die beschrijving van DDU2 volgens I4 ook een beschrijving van DDU2 volgens I3?

8.12. Beargumenteer alle "plussen" en "minnen" in Figuur 8.5 met behulp van de in de stof en in de opgaven gegeven eigenschappen van die voorbeelden.

□

AANGEHAALDE LITERATUUR

[ALM 82] F.W. Allen, M.E.S. Loomis en M.V. Mannino: "The integrated dictionary/directory system." ACM Computing Surveys **14**, 2 (1982), pp. 245-286.

[Ar 74] W.W. Armstrong: "Dependency structures of data base relationships." Proceedings IFIP congres, North-Holland, Amsterdam, 1974, pp. 580-583.

[BPZ 87] T.M.A. Bemelmans, J.A. van der Pool en N.J.M. Zwaneveld (eds.): "Poly-automatiseringszakboekje." Koninklijke PBNA, Arnhem, 1987.

[Br 80] E.O. de Brock: "Tables, table variables, and static integrity constraints." Memorandum, Technische Universiteit Eindhoven, 1980.

[Br 84] E.O. de Brock: "Database models and retrieval languages." Proefschrift, Technische Universiteit Eindhoven, 1984.

[Br 88] E.O. de Brock: "Some formal aspects of data dictionaries." Philips Research, NL-14.638, Eindhoven, 1988.

[CD 81] R. Curtice en E. Dieckman: "A survey of data dictionaries." Datamation **27**, 3 (1981), pp. 135-158.

[CO 71] CODASYL Data Base Task Group: "April 71 Report." ACM, New York, 1971.

[Co 72] E.F. Codd: "Further normalization of the data base relational model." In [Ru 72], pp. 33-64.

[Co 72b] E.F. Codd: "Relational completeness of data base sublanguages." In [Ru 72], pp. 65-98.

[Co 74] E.F. Codd: "Recent investigations into relational data base systems." Proceedings IFIP congres, North-Holland, Amsterdam, 1974, pp. 1017-1021.

[CP 87] S. Ceri en G. Pelagatti: "Distributed databases: Principles and systems." McGraw-Hill, New York, 1987.

[Da 86] C.J. Date: "An introduction to database systems (Vol. I)." Addison-Wesley, Reading (Mass.), 1986.

[Da 86b] C.J. Date: "Relational database: Selected writings." Addison-Wesley, Reading (Mass.), 1986.

[DDS 75] D. van Dalen, H.C. Doets en H.C.M. de Swart:
"Verzamelingen: naiëf, axiomatisch en toegepast."
Oosthoek, Scheltema en Holkema, Utrecht, 1975.

[DL 80] C. Delobel en W. Litwin (eds.): "Conference on distributed databases."
North-Holland, New York, 1980.

[GOV 84] G.E. Geurts-De Haas, H.P.F. van Oorschot, H. van Vondel e.a.:
"Ziekenhuisinformatiemodel."
Nationaal Ziekenhuis Instituut, 84.382, Utrecht, 1984.

[HK 87] R. Hull en R. King:
"Semantic database modeling: Survey, applications, and research issues."
ACM Computing Surveys **19**, 3 (1987), pp. 201-260.

[IDT 88] Independent Database Team - Holland: "Relational database in practice."
Proceedings IDT/C&D-conferentie, Londen, 1988.

[LP 82] B.W. Leong-Hong en B.K. Plagman:
"Data dictionary/directory systems: Administration, implementation, and usage."
Wiley, 1982.

[Mai 83] D. Maier: "The theory of relational databases."
Pitman, Londen, 1983.

[Ma 84] A. Mayne: "Data dictionary systems: A technical review."
NCC Publications, 1984.

[NGI 87] Databaseclub NGI:
"Het praktische gebruik van relationele systemen gerepresenteerd."
Proceedings NGI/SIC (C27), Utrecht, 1987.

[NGI 88] Databaseclub NGI:
"Het praktische gebruik van relationele systemen gerepresenteerd."
Proceedings NGI/SIC (C34), Amsterdam, 1988.

[Ol 78] T.W. Olle: "The CODASYL approach to database management."
Wiley and Sons, New York, 1978.

[Re 82] F. Remmen: "Databases: grondslagen voor de logische structuur."
Academic Service, Den Haag, 1982.

[RKS 88] M.A. Roth, H.F. Korth en A. Silberschatz:
"Extended algebra and calculus for nested relational databases."
ACM Transactions on Database Systems **13**, 4 (1988), pp. 389-417.

[Ru 72] R. Rustin (ed.): "Data base systems."
Prentice-Hall, Englewood Cliffs, 1972.

[Sh 81] D.W. Shipman: "The functional data model and the data language DAPLEX."
ACM Transactions on Database Systems **6**, 1 (1981), pp. 140-173.

[SM 80] F. Schreiber en G. Martella: "A data dictionary for distributed databases."
In [DL 80], pp. 17-33.

[SQL 89] ISO/IEC JTC1/SC21: "Database language SQL2."
ISO/IEC JTC1/SC21 N3155, draft proposal, 1989.

[TK 78] D.C. Tsichritzis en A. Klug (eds.):
"The ANSI/X3/SPARC DBMS framework: Report of the study group on data base management systems."
Information Systems **3**, 1978.

[Uh 73] P.P. Uhrowczik: "Data dictionary/directories."
IBM Systems Journal **12**, 4 (1973), pp. 332-350.

[Ul 82] J.D. Ullman: "Principles of database systems."
Computer Science Press, Rockville (Md.), 1982.

[We 86] C.J. Wertz: "The data dictionary: Concepts and uses."
North-Holland, Amsterdam, 1986.

[Ya 86] C.C. Yang: "Relational databases."
Prentice-Hall, Englewood Cliffs, 1986.

INDEX VAN DEFINITIES

0.13	$g \circ f$	$\{(x\,;g(f(x))) \mid x \in \mathrm{dom}(f)$ en $f(x) \in \mathrm{dom}(g)\}$	11
0.14 (a)	$f \upharpoonright B$	$\{(x\,;y) \in f \mid x \in B\}$	12
(b)	$f \not\upharpoonright B$	$\{(x\,;y) \in f \mid x \notin B\}$	12
0.15	$f \equiv g$ op A	$f \upharpoonright A = g \upharpoonright A$	13
0.16	$f \theta g$	$(f \not\upharpoonright \mathrm{dom}(g)) \cup g$	13
0.17	F is een verzamelingsfunctie	F is een functie en $\forall x \in \mathrm{dom}(F) : F(x)$ is een verzameling	15
0.18 (a)	$\Pi(F)$	$\{f \mid f$ is een functie over $\mathrm{dom}(F)$ en $\forall x \in \mathrm{dom}(f) : f(x) \in F(x)\}$	16
(b)	$\tilde{\Pi}(F)$	$\{f \mid f$ is een functie uit $\mathrm{dom}(F)$ en $\forall x \in \mathrm{dom}(f) : f(x) \in F(x)\}$	16
0.19	$\mathrm{He}(V)$	$\bigcup \{\mathrm{dom}(f) \mid f \in V\}$	16
0.20	r is een rij ter lengte n	r is een functie en $\mathrm{dom}(r) = [0 \,..\, n)$	17
0.21	$R \uparrow n$	$\{(r(0)\,; r(n)) \mid r$ is een rij ter lengte $n+1$ en $\forall i \in [0 \,..\, n) : (r(i)\,; r(i+1)) \in R\}$	18
0.22 (a)	$\mathrm{Tcl}(R)$	$\bigcup \{R \uparrow n \mid n \in \mathbb{N} - \{0\}\}$	19
(b)	R is cyclisch	$\exists (x\,;y) \in \mathrm{Tcl}(R) : x = y$	19
(c)	R is acyclisch	$\forall (x\,;y) \in \mathrm{Tcl}(R) : x \neq y$	19
1.1	T is een tabel over A	T is een verzameling en $\forall t \in T : t$ is een functie over A	21
1.2	v is een DB-toestand over g	v is een functie over $\mathrm{dom}(g)$ en $\forall E \in \mathrm{dom}(g) : v(E)$ is een tabel over $g(E)$	24
1.3	U is een DB-universum over g	U is een verzameling DB–toestanden over g	25
2.1	$T \upharpoonright B$	$\{t \upharpoonright B \mid t \in T\}$	30
2.2	$T \bowtie T'$	$\{t \cup t' \mid t \in T$ en $t' \in T'$ en $t \cup t'$ is een functie$\}$	31
2.3	$T \infty h$	$\{t \circ h \mid t \in T\}$	34
2.4	$B \to C$ in T	$\forall t \in T : \forall t' \in T$: als $t \upharpoonright B = t' \upharpoonright B$ dan $t \upharpoonright C = t' \upharpoonright C$	40
2.5	B is u.i. in T	$\forall t \in T : \forall t' \in T$: als $t \upharpoonright B = t' \upharpoonright B$ dan $t = t'$	45
2.6	B is minimaal u.i. in T	B is u.i. in T en $\forall B' \subset B : B'$ is niet u.i. in T	45

2.7		h verbindt T met T'	$T \upharpoonright \mathrm{dom}(h) \subseteq T' \infty h$	49
2.8		h verbindt T bilateraal met T'	$T \upharpoonright \mathrm{dom}(h) = T' \infty h$	49
2.9		$B \hat{\rightarrow} C$ in W	$\forall T \in W : B \rightarrow C$ in T	55
2.10		B is een sleutel van W	B is een verzameling en $\forall T \in W : B$ is u.i. in T	58
2.11		B is een minimale sleutel van W	B is een sleutel van W en $\forall B' \subset B : B'$ is geen sleutel van W	58
2.12	(a)	Head(W)	He($\bigcup W$)	59
	(b)	Ms(W)	$\{B \subseteq$ Head(W) $\mid B$ is een minimale sleutel van $W\}$	59
2.13	(a)	Prim(W)	$\bigcup$ Ms(W)	59
	(b)	Sec(W)	Head(W) – Prim(W)	59
2.14		W is in BCNF	$\forall B \subseteq$ Head(W) : $\forall a \in$ Head(W): als $B \hat{\rightarrow} \{a\}$ in W en $a \notin B$ dan is B een sleutel van W	61
2.15		W is in 3NF	$\forall B \subseteq$ Head(W) : $\forall a \in$ Sec(W): als $B \hat{\rightarrow} \{a\}$ in W en $a \notin B$ dan is B een sleutel van W	62
2.16		Un(E,U)	$\{v(E) \mid v \in U\}$	69
2.17		$B \hat{\rightarrow} C$ in E van U	$B \hat{\rightarrow} C$ in Un(E,U)	69
2.18	(a)	B is een sleutel van E in U	B is een sleutel van Un(E,U)	69
	(b)	B is een minimale sleutel van E in U	B is een minimale sleutel van Un(E,U)	69
2.19	(a)	Head(E,U)	Head(Un(E,U))	69
	(b)	Ms(E,U)	Ms(Un(E,U))	69
	(c)	Prim(E,U)	Prim(Un(E,U))	69
	(d)	Sec(E,U)	Sec(Un(E,U))	69
2.20	(a)	U is in BCNV	$\forall E \in$ He(U) : Un(E,U) is in BCNF	74
	(b)	U is in 3NV	$\forall E \in$ He(U) : Un(E,U) is in 3NF	74
2.21	(a)	H verbindt M permanent met D in U	$\forall v \in U : h$ verbindt $v(M)$	75
	(b)	H verbindt M permanent bilateraal met D in U	$\forall v \in U : h$ verbindt $v(M)$ bilateraal met $v(D)$	75
2.22		H is een DB-functie over U m.b.t. $(M\,;D)$	H is een functie over U en $\forall v \in U : H(v) \in v(M) \twoheadrightarrow v(D)$	76
2.23		$G \odot H$	$\lambda x \in \mathrm{dom}(G) \cap \mathrm{dom}(H) : G(x) \circ H(x)$	77

5.1 (a)	p is een transitie binnen U	$p \in U \times U$	141
(b)	R is een transitierelatie op U	$R \subseteq U \times U$	141
5.2 (a)	$(x\,;y)$ is reversibel in R	$(y\,;x) \in \mathrm{Tcl}(R)$	143
(b)	$(x\,;y)$ is irreversibel in R	$(y\,;x) \notin \mathrm{Tcl}(R)$	143
5.3	R is reversibel	$\forall p \in R : p$ is reversibel in R	143
5.4	$B \tilde{\rightarrow} C$ bij $(T\,;T')$	$\forall t \in T : \forall t' \in T'$: als $t \upharpoonright B = t' \upharpoonright B$ dan $t \upharpoonright C = t' \upharpoonright C$	146
5.5	$B \tilde{\rightarrow} C$ in E bij R	$\forall (v\,;v') \in R : B \tilde{\rightarrow} C$ bij $(v(E)\,;v'(E))$	147
6.1	q is een query over U	q is een functie over U	159
6.2	p is een view op U	p is een geordend paar en $\pi_2(p)$ is een query over U	178
6.3	V is een viewstelsel op U	V is een functie en $\forall p \in V : p$ is een view op U	178
7.1	f is een transactie op U	$f \in U \rightarrowtriangle U$	183
7.2	f is een transactie binnen R	f is een functie en $f \subseteq R$	185
7.3	$\mathrm{Ad}(U,R,f)$	$\mathrm{id}(U)\ \theta\ (R \cap f)$	185
7.4	$\mathrm{Adap}(U,f)$	$\mathrm{Ad}(U,\ U \times U, f)$	186
7.5	$\mathrm{Main}(U,R,E,\tau)$	$\mathrm{Ad}(U,\ R,\ \lambda v \in U : v\ \theta\ \{(E\,;\tau(v))\})$	194
7.6	$\mathrm{Inst}(U,R,E,t)$	$\mathrm{Main}(U,\ R,\ E,\ \lambda v \in U : v(E) \cup \{t\})$	195
7.7	$\mathrm{Insq}(U,R,E,q)$	$\mathrm{Main}(U,\ R,\ E,\ \lambda v \in U : v(E) \cup q(v))$	195
7.8	$\mathrm{Del}(U,R,E,q)$	$\mathrm{Main}(U,\ R,\ E,\ \lambda v \in U : v(E) - q(v))$	196
7.9	$\mathrm{Upd}(U,R,E,q)$	$\mathrm{Main}(U,\ R,\ E,\ \lambda v \in U : (v(E) - \mathrm{dom}(v))) \cup \{t\ \theta\ q(v)\,(t) \mid t \in \mathrm{dom}(q(v))\})$	198
8.1	I is een DD-functie	I is een functie en $\forall d \in \mathrm{dom}(F) : F(d)$ is een DB-universum	205
8.2 (a)	I is ten minste even expressief als I'	$\mathrm{rng}(I') \subseteq \mathrm{rng}(I)$	207
(b)	I is DD-equivalent met I'	$\mathrm{rng}(I') = \mathrm{rng}(I)$	207
8.3	I is DB-vormig	$\mathrm{dom}(I)$ is een DB-universum	210
8.4	I is zelfregistreerbaar	$\mathrm{dom}(I) \in \mathrm{rng}(I)$	210

INDEX VAN LEMMA'S, STELLINGEN, VOORBEELDEN, FIGUREN EN OPGAVEN

Met het oog op de veelvuldige verwijzingen geven we in bijgaand overzicht van alle lemma's, stellingen, voorbeelden, (genummerde) figuren en opgaven de (begin)pagina aan.

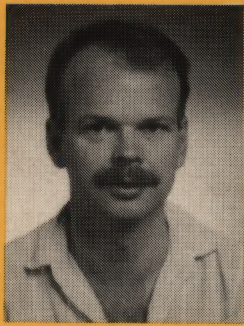
Opgaven

REGISTER

W

Z

Over de auteur



Bert de Brock is sinds eind zeventiger jaren werkzaam op het gebied van de theorie van databases en van de toepassing van deze theorie op praktijkproblemen.
Na zijn promotie op het gebied van databasemodellen en vraagtalen raakte hij ook betrokken bij de ontwikkeling van expertsystemen en kantoorautomatiseringssystemen.
Momenteel is hij als wetenschappelijk medewerker verbonden aan het Natuurkundig Laboratorium van Philips.
Hij is ook de ontwerper van de bekende Ziekenhuis-case die zowel door de Databaseclub van het NGI als door Codd & Date Ltd. werd gebruikt voor de evaluatie van de functionele eigenschappen van diverse bekende DBMS-en. Deze 'functional benchmark', die zich intussen mag verheugen in een warme internationale belangstelling, is in dit boek opgenomen.

Over het boek

De complexiteit van informatiesystemen neemt in de praktijk hand over hand toe. Het op formele wijze kunnen modelleren van de relevante gegevens voor een organisatie en het formeel specificeren van de operaties op deze gegevens wordt daarmee van cruciaal belang. Dit boek leert hoe dergelijke (systeemonafhankelijke) formele specificaties er uit kunnen zien en ook in complexe situaties praktisch bruikbaar zijn, en hoe aldus een belangrijk deel van de **semantiek** van de gegevens kan worden weerspiegeld in het formele model van een organisatie.

Het boek bevat meer dan 200 opgaven. Hierbij komen zowel theoretische als praktische aspecten aan bod. Veel aandacht wordt besteed aan het kunnen vertalen van informele omschrijvingen naar formele specificaties, en omgekeerd.

Het boek behandelt onder andere de volgende onderwerpen:

- wiskundige basisbegrippen,
- tabellen, databasetoestanden en database-universa,
- operaties op tabellen (projectie, join, renaming, etcetera),
- afhankelijkheden, (minimale) sleutels, normaalvormen,
- referentiële integriteit en databasefuncties,
- operaties op database-universa,
- generalisatie en specialisatie,
- classificatie van constraints,
- het systematisch construeren van database-universa,
- de Ziekenhuis-case,
- dynamische constraints en transitierelaties,
- raadpleging,
- transacties, en last but not least
- data dictionaries.

ISBN 90 6233 333 8
NUGI 852